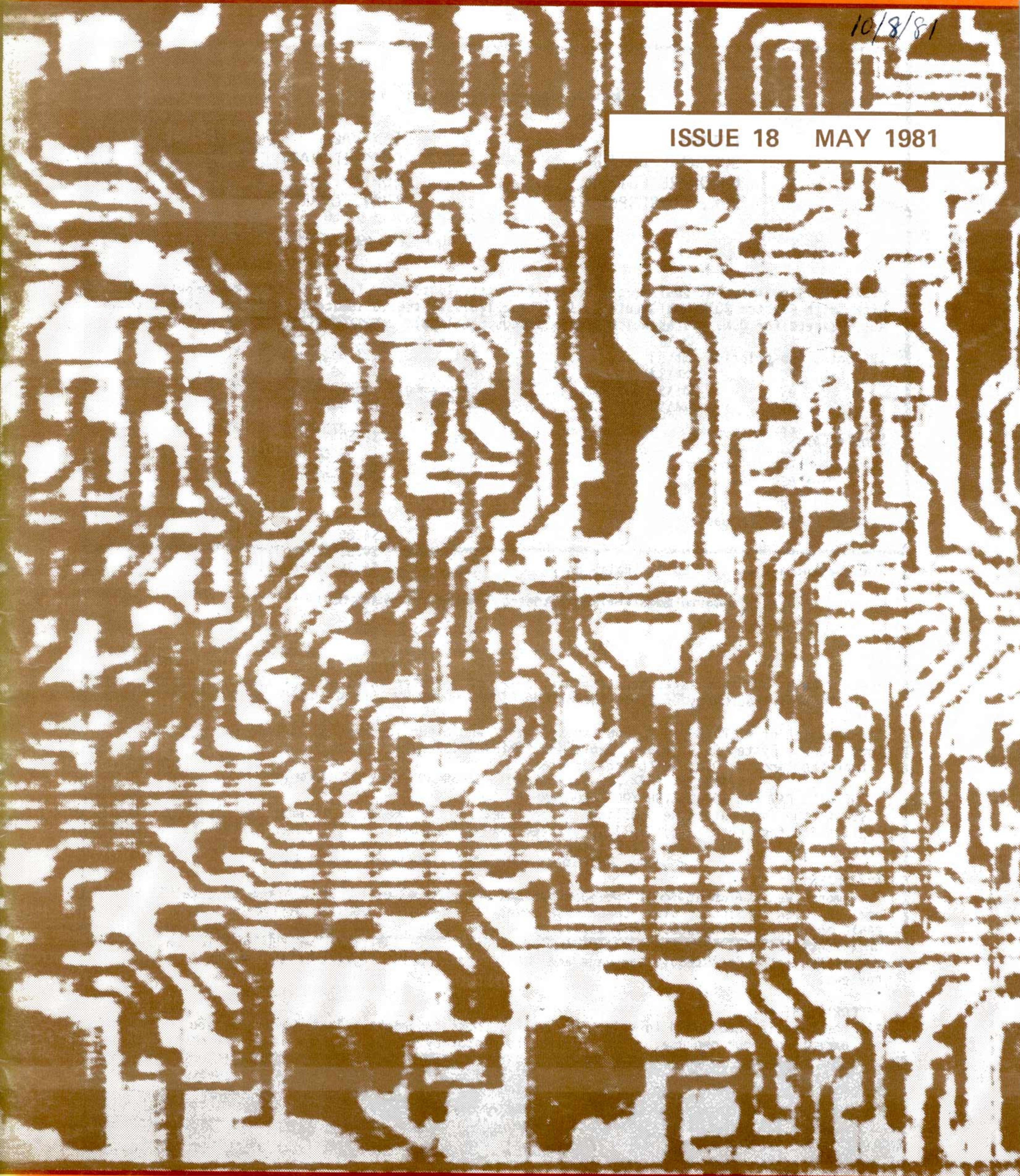


# TRS-80 SYSTEM 80 VIDEO GENIE

10/8/81

ISSUE 18 MAY 1981



# MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034, AUSTRALIA. TELEPHONE (08) 272 0966. PRICE: AUS. \$2.50, NZ. 3.00, U.K. £1.50

\*\*\*\*\* ABOUT MICRO-80 \*\*\*\*\*

EDITOR:	IAN VAGG
SOFTWARE EDITOR:	CHARLIE BARTLETT
HARDWARE EDITOR:	EDWIN PAAY
U.K. CORRESPONDENT:	TONY EDWARDS

MICRO-80 is an international magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80/Video Genie. It is available at the following prices (all prices shown in Aus.\$ except for U.K. prices which are in pounds Sterling).

12 months subscription	Aus.	\$24.00
	N.Z.	\$36.00 (Airmail)
	Hong Kong	\$46.00 (Airmail)
	U.K.	£16.00
Single Copy	Aus.	\$2.50
	N.Z.	\$3.50 (Airmail)
	Hong Kong	\$4.25 (Airmail)
	U.K.	£1.50
Months programs on cassette (at present available from Australia only)	Aus.	\$3.50
	N.Z.	\$4.00 (Airmail)
	Hong Kong	\$4.50 (Airmail)
	U.K.	\$4.75 (Airmail)
12 months subscription to magazine and cassette	Aus.	\$60.00
	N.Z.	\$78.00 (Airmail)
	Hong Kong	\$88.00 (Airmail)
	U.K.	£41.00 (Airmail)

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80 or Video Genie and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

\*\* WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS \*\*

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

\*\* CONTENT \*\*

Each month we publish at least one applications program in Level I BASIC, one in Level II BASIC and one in DISK BASIC (or disk compatible Level II). We also publish Utility programs in Level II BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

\*\* COPYRIGHT \*\*

All the material published in this magazine is under copyright. That means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

\*\* LIABILITY \*\*

The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

## \*\*\*\*\* CONTENTS \*\*\*\*\*

	PAGE
EDITORIAL	2
BETTER BASIC PROGRAMMING - PART 2	2
ESF FORUM	8
'80 USERS' GROUPS	9
INPUT/OUTPUT - (letters to the Editor)	10
MARKET PLACE	12
USING TRS-80 SOUND PROGRAMS ON A SYSTEM 80	13
READERS' REQUESTS	14
<u>SOFTWARE SECTION</u>	
BREAKOUT.....	L1/4K
INTERCEPT.....	L1/4K
▷ PRIME GENERATOR.....	L2/m1
▷ CHORD PRACTICE.....	L2/4K
▷ DOG RACE.....	L2/4K
▷ LIST DISABLE.....	L2/4K
SHARE TRANSACTION COSTS.....	L2/4K
▷ SHARE CHARTING.....	L2/16K
▷ PSYCHIC MASTER.....	L2/16K
MICRO-80 PRODUCTS	17
NEXT MONTH'S ISSUE	35
CASSETTE EDITION INDEX	36
ORDER FORM	36

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR:

MICRO-80, P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034. TEL.(08) 211 7244

U.K. SUBSCRIPTION DEPT:

24 WOODHILL PARK, PEMBURY, TUNBRIDGE WELLS, KENT TN2 4NW.

Printed by:

Shovel & Bull Printers, 312A Unley Road, HYDE PARK, S.A. 5061.

Published in Australia by MICRO-80, 433 Morphett Street, ADELAIDE.

★ ★ ★ FREE SOFTWARE OFFER ★ ★ ★

EVERY NEW AUSTRALASIAN SUBSCRIBER TO MICRO-80 WILL RECEIVE A FREE CASSETTE CONTAINING THREE LEVEL I AND THREE LEVEL II PROGRAMS PLUS COMPREHENSIVE DOCUMENTATION ... THE RETAIL VALUE OF THE SOFTWARE WOULD EXCEED THE COST OF THE SUBSCRIPTION!!!

## \*\*\*\*\* EDITORIAL \*\*\*\*\*

From the response we have had so far, it seems we struck the jackpot last month with our program listings using the normal size, emphasised print mode on the MX-80 printer. A number of readers have written commending this presentation so that will be our standard format from now on. We have made one slight adjustment this month and that is that we have used a special machine language program developed by Eddy Paay to control the line length of the listing to be exactly the same as on your screen, 64 characters. So what you see in print is what you will see on the screen.

A number of readers have requested that we illustrate as many of our programs as possible by showing the screen presentation they can expect to see. We have started doing so this issue via NEWDOS 80's screen dump facility and the MX-80's ability to print the 80's graphics set. Space permitting, we will do this more often in future issues.

Some time ago, we predicted that Tandy would discontinue sales of the Model I TRS-80 in Australia. We expected that it would have happened before now but a cancelled order by a large OEM left Tandy with considerable stocks. These are currently being cleared out at bargain basement prices, and it will not be long before our prediction will come true. We suppose that all good things must come to an end but we cannot help feeling sad at the passing of such a remarkable machine. We, and many others around the world, will continue to support the Model I for a long time to come so owners should not feel let down. Meanwhile, the Model III is shaping up as a worthy successor to the Model I and continues to impress us as we become more familiar with it. The earlier fears about software incompatibility have proven to be a little exaggerated, at least as far as BASIC programs are concerned. The refinements made over the Model I are all worthwhile and we are particularly impressed with the performance of the disk version, especially our own 1.4 megabyte system using MPI B92 disk drives.

The tradition of the Model I, of course, lives on in the System 80/Video Genie. From the U.K. we hear that a colour version of the Video Genie is imminent, whilst in Australia the business version has been available for several months. In the business version, the on-board cassette deck has been replaced by a numeric keypad, a lower-case display is fitted and an extra 2K of ROM has been installed in the previously unused area of memory addresses immediately above the BASIC interpreter. This ROM includes the lower-case driver routines and serial port driver routines. The business version of the System 80 is primarily intended to be used in disk systems for which the existing expansion interface is required.

Red Faces Department. Some little time ago, we received a letter from a reader containing draft manuscripts of the first three articles in a proposed series of about 10 articles, giving a guided tour of the '80's system RAM via the use of PEEK and POKE statements. We were so impressed that we put the letter in a safe place for an early reply which would have requested him to submit the complete series for publication. Yes, you've guessed what happened. That place was so safe that we still haven't found the letter! So please, inadvertently anonymous author, get in touch with us again so that we can bring your interesting series to all our readers.

- 0000000000 -

## \*\*\*\*\* BETTER BASIC PROGRAMMING - PART 2 - by Rod Stevenson \*\*\*\*\*

## \*\*\*\*\* ARRAYS \*\*\*\*\*

## INTRODUCTION

Many are frightened off by the mere mention of arrays and while we can truthfully say there is no need to be concerned, we also see this will not help. So therefore we hope this article will go a long way to making these rather useful functions more widely used.

We must admit though, we tend to use strings where others use arrays - this because of the very powerful string functions in our version of BASIC; other versions do not have such possibilities. But they do have more intensive array functions. Infinite BASIC will provide more of both string functions and array functions at a price of \$49.95. This is, after all, a personal preference as often the job can be done equally by string or array; one's ability with one or the other tends to be the deciding factor.

During discussion at the Adelaide Users' Group following one of our talks on this very subject of arrays, there emerged confirmation of a theory we had fuzzily held for some time. This is that the only difficulty with arrays is the proper understanding of the principle. Once this really is understood we can say, with absolute confidence, that you will never have any difficulty. Unfortunately, though, we cannot just give this understanding. All we can do is explain our way of thinking, and suggest you spend time thinking along the same lines until it really "clicks"; and it will. It's as though there is a "threshold" to pass - once over, no-one sees why there was ever a problem. How long did you take to work a telephone book which is, after all, a two dimensional array?

## HOW ARRAYS ARE STORED

An example of the need for true understanding is that you must really know the way you are going to need to access and use the data before you put it in. In other words, you must get it out the same way it went in. For instance, if you want to output to a printer in tabular form, you'll need the greater dimension to be vertical, for the practical reason that it will better fit on the paper.

However, all this business of numbers of dimensions, x, y, z directions, etc. is for our human way of thinking. The computer actually stores the data which composes the array in the same way it stores any other data. It doesn't really care that the human operator has chosen to have an array - there is only the one way the computer can store data anyway. Certainly (for example) the operating system keeps pointers to tell it that the operator will expect the data in location 27643 to be the nth element in an x dimensional array which has been dimensioned to z in each of its dimensions, and will accept alphanumeric data. These pointers could, however, have indicated that the data in that same memory location is an integer single precision number.

So you see the truth of our previous assertion that it really is only a problem (if there is one) of human understanding to make the fullest use of the array capabilities of the Level II BASIC.

## PRELIMINARIES

As explained in the manual, and in almost any text on arrays with BASIC, it is necessary to dimension the array to whatever size you wish to use it. This means both number of dimensions and number of elements in each dimension. At this point, you will also need to decide what variable name to assign to the array - and keep to it! For example, DIM\$(21,45) will give an array called A, which will accept alphanumeric data in two dimensions, with 21 pieces in the first dimension and 45 pieces in the second. Yes, we know the "wise ones" will exclaim that arrays start at 0, but we prefer to have you ignore this until you are completely confident to overcome the "stupidly simple" irritations of complexity this can cause. True, it is a waste of memory, but so what?

The default (that means, if you don't do it) is 10 in two dimensions and we must admit that we don't bother to dimension if it is less than 10 - though again, "wise ones" will despair at the blatant waste of memory. We do, however, dimension if there is more than one dimension. Beware of using up more memory than you have by dimensioning too many dimensions too big. Remember - A(20,20,20) will take up 8000 bytes in addition to the overhead of 12 bytes required by all arrays.

As with normal strings, space must be CLEARed for a work area big enough to contain the total expected array elements plus the longest one again. This CLEARing should be done before DIM, right at the start of your program.

Any of the normal variable names can be used as an array name. You can define them as integer, single precision, double precision, as needed, and again, memory will be saved. But again, we usually omit anything that seems unnecessary. Yet again in this article, as in the last one, may we suggest the use of meaningful variable names as array names. It is just a way of thinking, but it does seem to us logical to have an array called D if it will contain rates of discount allowed.

## ACCESSING ARRAYS

If there is anything difficult about the use of arrays, it is that they can be accessed in so many ways. It is when accessing arrays (whether to read or write) that the importance of the idea at the start of this article applies: think about the whole concept of an array until you are totally at ease with it. Obviously, whatever way data is put into an array for storage is the way it must be got out. Too often this simple point is overlooked - we can only assume through a lack of basic understanding of the mechanism. Remember that we said that the important thing is how the human operator thinks of his array - the computer doesn't really care, as it stores the data the same way regardless.

Now to details. The name you choose for the array variable is the one it is stored under. It must remain the same from start to finish. However, the elements in the array (called subscripts) will have numbers allocated to them by the operating system (see Diagram 1, left), and will always start from 1 (actually 0, but as we said earlier, disregard this) and proceed in numerical sequence for however many elements you need to have.

This is probably what makes arrays so useful and easy to access - instead of having a value stored in a series of different variable names, lots of pieces of data can be stored under the same variable name, just being identified by their own individual subscript. Rather like a family all living under one roof but still being total individuals.

DIM A(10)

A(1)
A(2)
A(3)
A(4)
A(5)
A(6)
A(7)
A(8)
A(9)
A(10)

So, any slot in the array can be accessed by using a variable as the subscript value. If X = 3, then A(X) is exactly equal to A(3). What really stands out here is the extreme suitability of a FOR...NEXT loop.

### 1. For input.

```
10 FOR I = 1 to 10
20 PRINT "VALUE FOR ELEMENT" I; : INPUT A(I)
30 NEXT I
```

Or instead of input from a keyboard, have the array filled by DATA lines.

```
10 FOR I = 1 to 10
20 READ D:A(I)=D
30 NEXT I
```

### 2. For output.

```
10 FOR I = 1 TO 10
20 PRINT "VALUE NUMBER"; I; "IS" A(I)
30 NEXT I
```

However, individual elements can be accessed just as simply by the use of a variable to specify the subscript.

```
10 INPUT "DISCOUNT CODE"; I
20 PRINT "TOTAL IS" T-(T*A(I))/100
```

Both of these last examples are extreme in their simplicity. The array may merely be the end storage place for the result of a long and complex calculation, and it may be the source of the initial data to start off the same long and complex calculation. It can even be contained within a large FOR...NEXT loop with the end result being replaced where it initially came from.

## ONE DIMENSIONAL ARRAYS

These are perhaps the most common, also called vectors. There really is little difference between this and a string. The only practical point is that arrays are already provided with computer-made subscripts, whereas the user must furnish them to use a string. In fact, we use strings rather than one-dimensional arrays. There is also a memory-saving but this is not significant enough to worry about.

The uses of one-dimensional arrays are so numerous that it is hard to imagine the total possibilities. But in reality, all of these uses are based on the fact that the array is a ready-made table of data.

As mentioned above, a table of discounts can be an array to be accessed by an input number or even by another array or look-up table. This last is a little advanced to cover yet.

As a look-up table for input checking, the array is particularly suitable. More about this in the instalment on checking input.

Perhaps a practical example of checking data using an array. This program is a simple number generator for cross lotto games. It uses an array to check if the current number generated by RND has previously been generated. Clearly, this is a rather simplistic example. It does show how to check with an array, however. Of course, the calculation to arrive at the data could well be more complex than the simple RND function, but the principle is the same. The array need not contain all zeros either - could well be some form of data check system.

```

100 DIM A(40)    'DIMENSIONS THE LOOK-UP ARRAY TO 40
110 FOR J = 1 TO 8  '8 SETS OF NUMBERS
120 FOR I = 1 TO 40 : A(I) = 0 : NEXT I  'SETS EACH ELEMENT IN THE
                                         'ARRAY TO ZERO
130 FOR I = 1 TO 6  '6 NUMBERS IN EACH SET
140 X = RND(40)    'GENERATE A RANDOM NUMBER
150 IF A(X) <> 0 THEN GOTO 140  'CHECKS IF ARRAY ELEMENT EQUAL
                               TO RANDOM NUMBER IS 0. IF NOT
                               THE NUMBER HAS BEEN PREVIOUSLY
                               GENERATED.
160 A(X) = X      'ARRAY ELEMENT EQUAL TO NUMBER GENERATED NO LONGER
                               EQUAL TO ZERO
170 PRINT USING "#####";X;  'PRINT USING FORMATS OUTPUT. MORE ON
                               THIS IN THE LATER ARTICLE ON IT
180 NEXT I      'NEXT NUMBER IN SET
190 PRINT      'SEPARATE LINES FOR EACH SET
200 NEXT J      'ANOTHER SET OF 6

```

To save a plethora of letters which we are sure would otherwise be precipitated, perhaps a small divergence into the nature of random numbers.

The more technical aspects of this was the subject of a piece in the Adelaide Users' Group newsletter, and we don't really feel it's appropriate here. Suffice it to say that the RND function can be fooled into producing a non-random sequence by POKEing values at appropriate points in the program.

What we do want to repeat here is that if the cross lotto selection is truly random, there is no way to predict successive results. In fact, the sequence 1 2 3 4 5 6 could well be chosen at random. Of course, we as humans would strongly doubt that it is random: it is. Random simply means that any number has as much chance as any other of selection. Clearly, the fact that its preceding-numbered neighbour was just picked should in no way influence that equal chance. Nor should the fact that it has been chosen six preceding times - each choice is an event of its own with no relationship to any other choice.

Over a large period of selections one would expect each number to be selected an equal number of times. By a large period of selections we really do mean large - millions. To test this theory a simple program to tabulate the general-frequency of the RND selector will illustrate.

```

90 DIM A(40)    'DIMENSION ARRAY FOR COUNTING
100 FOR I=1 TO 40: A(I)=0:NEXT    'ZERO ARRAY BEFORE START
110 INPUT "HOW MANY"; N  'NUMBER OF NUMBERS TO GENERATE
120 FOR I=1 TO N    'DO IT N TIMES
130 X=RND(40):A(X)=A(X)+1:    'GENERATE A RANDOM NUMBER, COUNT IT
140 NEXT I      'DO IT AGAIN
150 FOR I=1 TO 40: PRINT I; A(I), :NEXT  'PRINT ARRAY OF COUNT
160 PRINT TAB(14) "TOTAL GENERATED ="N  'TOTAL NUMBER GENERATED
170 PRINT: GOTO 100   'RETURN FOR ANOTHER LOT

```

By experimentation, you will see the higher number you specify, the more equal will be the times each one is selected. You could also test the inclusion of the RANDOMIZE function by putting this in line 80.

#### TWO-DIMENSIONAL ARRAYS

It is here that some lose the ability to come to grips with arrays - also called a matrix. But there is really no need to. We do implore that if you haven't already done so, you stop now to truly dwell on the nature of multi-dimensional arrays.

The way we mentally picture it - remember we said earlier that it's how you imagine things that matters - is as a square made up of smaller squares with each box representing one array element. While this may seem simplistic we have had no program revolt on us; until we do, we will continue to think in this way.

DIM A(4,5)

A(1,1)	A(1,2)	A(1,3)	A(1,4)	A(1,5)
A(2,1)	A(2,2)	A(2,3)	A(2,4)	A(2,5)
A(3,1)	A(3,2)	A(3,3)	A(3,4)	A(3,5)
A(4,1)	A(4,2)	A(4,3)	A(4,4)	A(4,5)

The above diagram shows the earlier one-dimensional array which went vertically down the page, having had each row extended sideways into a second dimension.

The use of such two-dimensional arrays is also legion: wherever a second set of information depends on, and belongs with, a first set. The common telephone book is an obvious example - if the names were not present there would clearly be no need for numbers, and the numbers, of course, would be useless without names. However, there is a trap in thinking that the second dimension is limited to one piece of data, forgetting that just as the first dimension extends downwards, so the second one extends across. Consider a payroll array: employee number in the first (downwards) dimension, days of the week in the second (across) dimension.

#### MULTI-DIMENSIONAL ARRAYS

As far as your writer is concerned, three-and-more dimensions become a little academic. We can well understand their use in pure maths, three-dimensional plotting, multiple critical path decisions. Of this last use, the Tandy Visicalc program is an excellent example, though we do wonder at (and would be mentally quite unable to cope with) a 255 dimension array which is theoretically possible if the computer has enough memory. As far as optimum path choice goes, we find it simpler to use a set of FOR...NEXT loops, keeping only the pieces of data and decisions which lead to the best end result. However, we hope the reader can see that it would be easy enough to store each of these sub-results and decisions in arrays along the way.

In three-dimensional arrays, conventionally x is the first, y is the second, z is the third.

#### STRING ARRAYS

We must confess that we find these the most useful of all arrays - possibly following our partiality to strings.

The string array is exactly like the numeric array except that it has been defined as a string array and so is able to accept alpha as well as numeric data.

There is some possibility for confusion as to the length of each string capable of being stored in each element of the array. Following the example of numeric arrays, a casual thought may be that only one character can be held in each element. This is not so: while each numeric array element can hold only one number, this will take more than one byte (up to 8). Similarly, the string array can hold up to 255 characters in each element (providing string space is available, and you don't run out of memory).

The listed program will produce an array like this but, of course, without the lines which have been put in only to aid in the human way of thinking of an array as a series of boxes.

DIM A\$(4,5)

A\$(1,1) SMITH	A\$(1,2) TOM	A\$(1,3) NORWOOD	A\$(1,4) HOLDEN	A\$(1,5) 24
A\$(2,1) JONES	A\$(2,2) BILL	A\$(2,3) MILE-END	A\$(2,4) FORD	A\$(2,5) 43
A\$(3,1) BAKER	A\$(3,2) ALICE	A\$(3,3) ADELAIDE	A\$(3,4) DATSUN	A\$(3,5) 17
A\$(4,1) VAGG	A\$(4,2) DELPHINIA	A\$(4,3) SPRINGFIELD	A\$(4,4) MERCEDES	A\$(4,5) 36

```

90 CLEAR 300 :REM0N
100 DIM A$(4,5)  'SIZE OF EXAMPLE ARRAY
110 FOR I = 1 TO 4  'NEXTJ,I
120 FOR J = 1 TO 5  'Y DIMENSION
130 READ D$: A$(I,J)=D$  'PUT DATA INTO ARRAY
140 NEXTJ,I
150 'PRINT OUT ARRAY TO TEST IT
160 FOR I = 1 TO 4: FOR J = 1 TO I,J);";":NEXTJ:PRINT:NEXTI
200 DATASMITH,TOM,NORWOOD,HOLDEN,24
210 DATAJONES,BILL,MILE-END,FORD,43
220 DATABAKER,ALICE,ADELAIDE,DATSON,17
230 DATAVAGG,DELPHINIA,SPRINGFIELD,MERCEDES,32

```

This is merely a short example to illustrate the idea - if this is the total quantity of data an array is probably unnecessary. This array is to store data about people and, while it has only five elements in the second dimension, there can be as many as necessary to hold all the separate pieces of data required. The array can be filled by input or data lines. Data lines would seem more appropriate, as probably every time the program is run the array will hold the same data. Of course, this can still be changed by individual input statements which search for erroneous data, then correct it. The following program illustrates a simple case - the input method used is shown only for simplicity - in fact, we hope you will use one of the input checking routines to appear in the next instalment on input checking.

```

380 PRINT
390 PRINT
400 INPUT"NAME, SUBURB";N$,S$  'GET NAMES TO BE SEARCHED FOR
410 FOR I = 1 TO 4  'STEP THRU ALL X DIMENSIONS
420 IF A$(I,1)=N$ AND A$(I,3)=S$ THEN F=I: I=4:NEXTI:GOTO460  'IF MATCH FOUND
430 NEXTI
440 PRINT"NOT FOUND":GOTO400  'IF NO MATCH
460 FOR I = 1 TO 5:PRINTA$(F,I);";":NEXTI:PRINT  'SHOW WHAT FOUND
470 INPUT"ANY CHANGES";C$  'ALLOW FOR CHANGES
480 IF LEFT$(C$,1)<>"Y"THEN GOTO600  'IF NO CHANGES, CONTINUE
490 INPUT"NEW DATA (LAST NAME, FIRST NAME, CAR, AGE)";A$(F,1),A$(F,2),A$(F,3),A$
(F,4),A$(F,5)  'NEW DATA
500 PRINT:GOTO160  'SHOWS NEW DATA IN ARRAY

```

The above example is merely to stimulate thought - as indeed are all the examples, routines, and programs in this series. As we said right at the start, the best learning is gained by experimentation. So please try out our examples, then extend them to be useful to yourself.

While this particular example is only two-dimensional, it could just as easily be three-dimensional. Perhaps if there is another set of information which depends on the type of car - although this would be unnecessarily complicated we feel. It would also create another dimension "behind" all the sideways elements too. In this case, the way we would handle it is to use a longer string to contain the related information in the same element. To make searching easier, we suggest packing all items of the same nature out to the same length with blanks.

```

600 FOR I = 1 T'ADD ONE SPACE AT A TIME
610 FOR L=LEN(A$(I,0))+1 TO 12  'PACK EACH ELEMENT TO 12
620 A$(I,J)=A$(I,J)+" ";NEXTL  'ADD ONE SPACE AT A TIME
630 NEXTJ,I
635 ' PRINTOUT THE ARRAY IN ITS NEW FORMAT
640 PRINT:FOR I=1 TO 4: FOR J=1 TO 4: PRINT:A$(I,J);":NEXTI:NEXTJ

```

Such a technique as this will make each address to have a length of 12 characters even if the name is only short. Then, for instance, the post-code can be added to make a length of 16 by a simple string addition technique such as

A\$(I,3)=A\$(I,3)+PC\$.

Then having filled the array with a set of addresses containing post-codes as the last four characters of the address element, the post-codes can simply be retrieved by RIGHT\$(A\$(I,3),4) and sorted, compared with input, etc. Because the screen layout will be used up in its full width by the above example, we have not given an exact example of what we suggest, but we have given a similar example of using right string and we suggest that you look at this and work out for yourself what we have done.

```

730 FOR I=4 TO 1 STEP -1  'DO IT IN REVERSE ORDER FOR EFFECT!
740 PRINT TAB(30)RIGHT$(A$(I,3),6)  'SELECT LAST 6 CHARACTERS IN ARRAY ELEMENT 3
IN Y DIMENSION FOR EACH X DIMENSION
750 NEXT

```

## \*\*\*\*\* ESF FORUM - by Les Kinch \*\*\*\*\*

This month we have handed over the ESF FORUM to reader Les Kinch who has some handy advice to give concerning the use of the stringy floppy and also provides a list of parameters for saving common machine language programs onto wafer. There are two points to make arising from the contents of Les's article. At the time the article was written, Exatron was forced to change to a different tape formulation which was softer than earlier tapes and tended to leave considerable deposits on the head and capstan. Since then, Exatron has changed to a harder formulation and also improved the design of the wafer so that there is much less risk of tape damage. Nevertheless, Les's advice to clean the head and capstan frequently is still very important - remember, the tape is running more than five times faster than a standard cassette recorder.

When cleaning the head etc., use sparing quantities of isopropyl alcohol or a proprietary head cleaner which will NOT attack plastic. Acetone is NOT suitable. We have recently had several ESF's with 'Write Protected' errors returned for service. We found in each case that the LED which detects the presence of the write-enable sticker on the wafer had been etched to the point where it could no longer recognise the presence of the sticker. This was presumably caused by the use of excessive amounts of an unsuitable head cleaning fluid. Now, over to Les.

First off, if any of you are hesitant in making the plunge, forget it! The 'Stringy' is everything its designers say it is. A very fast input/output device. But, be prepared to give it more attention than you would to your old, slow, audio recorder. Remember it utilizes a similar principle to that used on the 'Floppy-disk' recorder and, in particular, is prone to contamination to the read/record head from the media itself. So, be prepared to clean your record head two to four times a week. This is a users' fact - (I use a proprietary brand of liquid head cleaner applied with cotton buds).

Exatron admit that some of the recording media will and do leave a deposit of oxide on your heads. They are right, it does! I've a fair knowledge of recording tape quality and consider that the 'Stringy' tape should be the best. In practice, it is not. This will keep the price down a bit but it does mean you will have to clean frequently.

You may think I've belaboured that point to excess, but I haven't really. For instance, you have a good program as No.1 on your 'Stringy'. You want to erase the rest of the tape ready for a new program. Type @NEW2. This should NEW the remainder of the Stringy Floppy. If you have a dirty head, this procedure may well 'NEW' the whole tape. Don't get me wrong - I'm not saying the Stringy Floppy is a bad buy, it's great, but be more careful.

Another good habit is to re-initialize the bootstrap each time you wish to use it. For instance, after you have put 15 plus K into your memory and you want to "@SAVE" it, go to the initialization routine first (System \*/12345) and then @SAVE your program.

I have included a short list of some of the programs I have currently changed over to my 'Stringy Floppies'. If you have problems in converting your favourite program to the media, please write to me - I'll do my best to help (not promising results of course, but I can try!)

PROGRAM	START	LENGTH	AUTO-START
MONBUG	28672	4096	28672
RSM 2	27648	4609	27648
PENCIL	21897	5900	23649
TR COPY	17152	1521	17152
SARGON II	18944	12196	20480
FASTGAMMON	22272	4736	17232
BARRICADE	17232	2736	17232
SAUCERS	20480	8192	20480
INVADE	17174	3244	17717
MONBUG (LOW)	17280	4096	17280
A.T.C.	17152	2617	17152
FORMUL	17408	1098	17408
LIFE 2	17664	582	17664
MICRO MUSIC	17152	1649	17152
MICRO MOVIE	17152	2560	17152
T-BUG	17280	1536	17280
ZBUG	17280	6063	17280
T-BUG (HIGH)	29936	1295	29952
BASIC I	28160	4607	28160
BASIC III	16896	5572	16896
CKRS 80	20480	10016	20480
RENUM	31808	959	31820
EDTASM	18056	5767	18056
EDTASM PLUS	17280	12002	17280
PYRAMID	17152	15510	17152
BMON 16	29248	3262	31641 MEM.SIZE 29200
LOADER	31796	0689	32051

<u>PROGRAM</u>	<u>START</u>	<u>LENGTH</u>	<u>AUTO-START</u>
PACKER	28060	4608	28060 MEM.SIZE 28059
FLIGHT SIMUL I	17136	11025	17136
FLIGHT SIMUL II	17136	11025	17136
ADVENT (COUNT)	17136	11025	17136
ADVENT	17136	11025	17136

- 0000000000 -

\*\*\*\*\* '80 USERS' GROUPS \*\*\*\*\*

The following is a list of '80 Users' Groups. If you have a group that is not included here, please let us know about it so that we can publish details. Owners of System '80s are welcome at all the groups.

\*\* AUSTRALIA \*\*

BRISBANE: Contact: Mr. Lance Lawes,  
Tel: Home (07)396 2998  
Bus: (07) 268 1191 Ext. 15  
MEETINGS 1st Sunday of the month at 2 p.m. at 21 Rodney St. Lindum.

MELBOURNE: NORTHERN AND WESTERN SUBURBS

Contact: Mr. David Coupe (03)370 9590  
MEETINGS: C.P.M. Data Systems, 284 Union Road, Moonee Ponds.  
Alternate Thursdays at 7 p.m.

EASTERN SUBURBS

Contact: Mr. John Fletcher, 89 0677 bet. 9-4.  
MEETINGS: 4th Wednesday of the month at 7 p.m. at Kingswood College,  
355 Station St. Box Hill.

FRANKSTON: PENINSULAR GROUP  
(VIC.) Contact: M.G. Thompson (03)772 2674  
MEETINGS: 2nd Tuesday of the month (except Jan.)

GEELONG: GEELONG COMPUTER CLUB

Contact: The Geelong Computer Club, P.O. Box 6, Geelong, Vic. 3220.  
MEETINGS: 2nd Tuesday of the month at TYBAR Engineering, Hampton  
St. Newton.

DARWIN: Contact: Tony Domigan, P.O. Box 39086, Winnellie, N.T. 5789.

ADELAIDE: Contact: Rod Stevenson, 51 5241 bet. 9-4. 36 Sturt Street, Adelaide.

CANBERRA: Contact: Bill Cushing, 10 Urambi Village, Kambah ACT. 2902.  
(PH. 31 6399).  
MEETINGS: 3rd Thursday of each month at 7.30 p.m. in:  
Urambi Village Community Centre, Crozier Circuit, Kambah.

\*\* UNITED KINGDOM \*\*

National Users Group in U.K.  
Brian Pain, 40 High Street, Stoney Stratford, Milton, Keynes.

International TRS-80 Level I User Group,  
Secretary: Mr. N. Rushton, 123 Roughwood Dr., Northwood, Kirkby,  
Merseyside, L33 9UG.

NEWCASTLE: Contact: John Stephen Bone 0632 770036  
NPCS (Newcastle Personal Computer Society)

\*\* NEW ZEALAND \*\*

AUCKLAND: Contact: Ron Feasy 799 366 (Bus:) 469 455 (Priv.)  
MEETINGS: 1st Tuesday of each month, 7.30 p.m. at:  
NZ Solenoid Co. Ltd.,  
28 Kalmia Street,  
Ellerslie, Auckland.

- 0000000000 -

## \*\*\*\*\* INPUT/OUTPUT \*\*\*\*\*

From: Ron Jorgensen, Vic.

Enclosed is a cheque for renewal of my subscription to MICRO-80. When the time came, the decision to renew my subscription was not as easy to make as I would have imagined last year after reading my first 4 or 5 issues (beginning with Issue 6). Lately, on reading my eagerly awaited copy, I have been a little disappointed that the excellent standard of those early issues has not been entirely upheld.

I realise that yours is primarily a software-oriented magazine (good) and indeed, the April and May 1981 issues, to take examples, included some interesting programs, but lacked elsewhere. The April issue included the usual Market Place, Readers' Requests, Users' Group list, the Editorial and no less than six reviews. After the just as noticeable lack of them this month, I hope that readers' reviews will return in moderation in Issue 19. I agree that readers' reviews are an important new part of the magazine, but some discretion must be used in their selection. Reviews of games could perhaps be shorter and more concise, whereas utilities that often warrant a more detailed treatment could be given it. I liked the "toolkit" and "editor" reviews.

The Editorial in April consisted of only 1 topic, whereas editorials of the past have included many subjects and titbits of information on varied subjects. We all like to hear what is happening in the microcomputer industry, especially those of your readers who do not go to computer club meetings. There must be a million things that you at MICRO-80 talk about - what is happening at Fort Worth, Rydalmer or Dick Smith; how is the PMC-80 and Video Genie distributed overseas; ROM differences and many other things - so why not share these with us, your readers? Certainly the old rumour-mill provoked much thought and conversation. If not in the Editorial, why not start a gossip/information called "Online" or something, as in the business computing section of the Melbourne "Age".

Sad was the loss of Peter Hartley, but more so the supposed demise of "Better Bytes" and his straightforward, frothy style. The varied information in "Better Bytes" was always welcome, ranging from programming to cassette care. "Better Basic" is a great new addition, and appears set to fill the gap left by "G.T. Basic". Everyone knows now, if not before, how to POKE out the BREAK key but there must be many more little tricks involved with poking around in that mystery-shrouded area labelled "reserved". So please, fellas, tell us about 'em!!

I have always thought that letters were important to any magazine, especially to one that has said it is very interactive. Where then, is Input/Output? It is always so very interesting to hear about other users' opinions and problems.

Finally, I would like to say that I really liked the complete and objective review of Model 3 - and the pictures were great. Incidentally, how many of your readers have Model 3's? Is a survey possible?

I hope that my thoughts here have been of the form of constructive criticism, and that they help to make MICRO-80 bigger and better than before, while still having that "club" type feel to it. Incidentally, not publishing the assembled code for the lower-case drivers is not the way to keep that atmosphere. I still love your magazine, and hope I always will. Thanks.

(Thank you for your helpful comments, Ron, many of which have been echoed by a number of other readers. There are several reasons for the changes upon which you comment. Many were unintentional. The loss of Peter Hartley caused a considerable hole in our "resident" editorial staff which we are still patching up and, coupled with a desire to experiment in response to a number of readers' requests, we included a number of reviews of software. In retrospect, we should have spread them out over several issues and perhaps edited some a little more thoroughly.)

Input/Output and some of the editorial comments disappeared temporarily because of the pressure of work on the editor associated with our move to new premises and the desire to catch up on publication dates, which is a constant worry to us. Incidentally, we have been answering many more readers directly in recent times, although we still have a considerable backlog of mail. By now, you will have seen the return of both these features.

Photographs (sigh). We held up the printing of the March issue by a good week in a vain attempt to obtain some photographs of the Bathurst race to adorn our front cover. Dismal failure - so, conscious of deadlines, we went to press with the old design and have done so ever since, until this issue. We will continue the use of "screen dumps" for illustration and the occasional photographs of equipment until we are at last publishing the magazine in the month stated on its cover. Then we will have more time to chase photographs, etc.

Ron, the reason we did not publish the assembled code for the lower-case drivers is that it is not possible to produce this code in any way other than with an editor/assembler. The code resides in screen RAM and would be interfered with by the monitor itself. We were certainly not trying to generate extra cash for MICRO-80 by selling the object code. In fact, the offer of the cassette for \$5.00 was an afterthought when we realised that not all readers would be able to use the program because of a lack of an editor/assembler.

Thank you for your comments, Ron (and those readers who have made similar comments). It is obvious that the formula we developed in our first 14 or so issues is the most acceptable to our readers and we will return substantially to that formula. - Ed.)

FROM: B.N. Hall - 49 Bastick Street, Rosny, Tasmania, 7018.

I was very interested to read the comments by Mr. A. Smallbridge (Dec. 80) regarding cassette loading of the System 80. Having had sporadic problems with mine and believing that level was the problem, I launched into the modifications involving a LED and installation of an accessible level control pot.

Dick Smith's instructions tell the user to adjust the positive peaks of the recovered signal to 3 volts and therein lies the rub. There is no way that tapes recorded on my machine (and apparently other System 80's) will produce anything like the symmetrical, almost sine wave, signal that Dick's demonstration tape and one from MICRO-80 will give.

I experimented with the erase head (DC) using no input and reversed input, demagnetised the heads and low pass filtered the input signal so that it approximated a 'good' output signal, all to no avail. The only waveform obtainable at the output was violently assymetrical with a large excursion in the negative direction and a couple of positive overshoots.

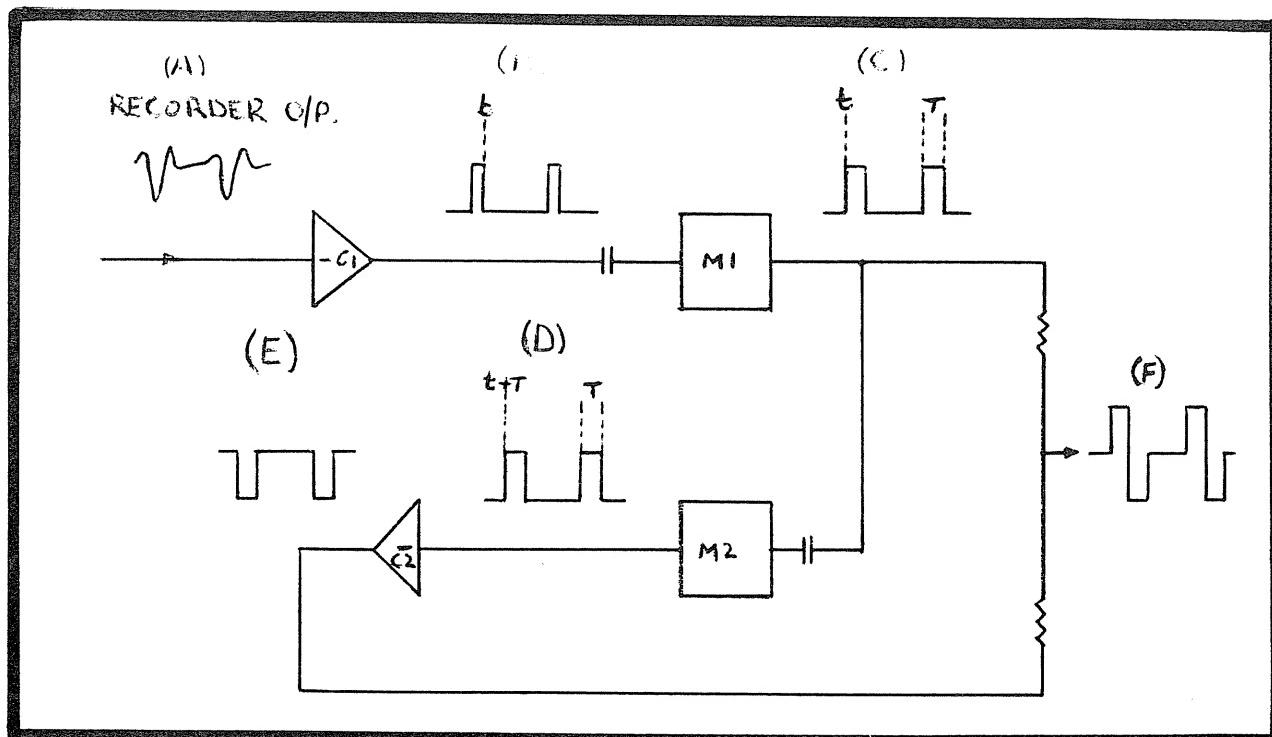
In desperation I built up some electronics which used the negative excursions as the trigger to derive an output signal from the recorder similar to that applied in the record mode. Lo and behold, no more loading problems.

The circuitry is relatively simple and uses a quad comparator (LM339, two spare) and a dual timer (NE556) used as two one-shots. A block diagram is attached.

The first comparator C1 produces an inverted pulse train (B) timed by the negative excursions of the signal from the recorder (A). The first one-shot M1 is triggered by the trailing edge of (B) and produces a positive going pulse train (C) whose pulse duration T, is about 120 microseconds. The second one-shot is triggered by the trailing edge of (C) and produces a similar waveform (D) delayed by T. (D) is inverted by comparator C2 whose output (E) is resistively combined with (C) to produce (F).

The device is powered from the 80's internal 9v rail and produces a signal of about 5v p-p with which the 80 seems quite happy.

I will be pleased to supply further details to any interested reader.



(Thank you very much for this contribution, Mr. Hall. If you are inundated with requests for detailed circuit diagrams and fitting instructions, perhaps you might like to prepare a detailed article for us to publish - Ed.)

## \*\*\*\*\* MARKET PLACE \*\*\*\*\*

Market Place is available to any reader who has hardware to dispose of. An entry costs nothing - you pay MICRO-80 \$5.00 or 5% commission, whichever is the greater - up to a maximum of \$30, after the goods are sold. The commission is calculated on your advertised price.

KTM-2 KEYBOARD UNIT in good order	\$235.00	Mr. A.R. Hall, c/o Post Office, Southbrook, Qld. 4352.
-----------------------------------	----------	--

\*\*\*\*\*

AXION SERIAL THERMAL PRINTER with 8 rolls of paper and TRS-232 interface	\$400.00	Mr. T. Domigan, P.O. Box 390986 Winnellie, N.T. 5789.
---	----------	---

\*\*\*\*\*

SYSTEM-80 LEVEL II 16K MEMORY, GAMES CASSETTES, BOOKS AND MANUALS, brand new	\$680.00	Mr. M. Gosbee Ph. (02) 88-4468
---	----------	-----------------------------------

\*\*\*\*\*

SYSTEM-80 L2/16K with full documentation, approx. 40 blank cassettes (data), books, magazines, and programs (mostly games) including EDTASM PLUS and Adventure No.9.	\$720.00 o.n.o.	Mr. R. Glucz, 1/1707 Dandenong Rd. Oakleigh, Vic. 3166 Ph.(03) 543 3302 A.H.
---	-----------------	---

\*\*\*\*\*

TRS-80 MODEL 1 Level II 4K excellent condition, Catalogue price \$879. Sell with manuals and programs	\$750.00	James Grigg. Ph. (052) 78 5056
---	----------	-----------------------------------

\*\*\*\*\*

Radio Shack QUICK PRINTER: 150 LPM on 4.75" wide aluminised paper. Software selectable 80, 40 or 20 characters per line. Upper-case with full descenders. Automatic underlining. Seldom used and in excellent working order. Catalogue price \$699. Will sell for:

\$400.00

K. Cook,  
P.O. Box 931,  
GOVE, N.T. 5797.

SYSTEM-80 L2/16K with over \$100 in programs,  
including BMON, RPN Calc. Sargon 3.5 Chess,  
Space Invaders, 70 Maths/Finance Programs.  
Excellent condition

\$680.00

G. Hobbes,  
42 Nareen Pde.  
NARABEEN, N.S.W. 2101.  
Ph. (02) 913 2264.

SERIAL PRINTER INTERFACE

\$ 40.00

\*\*\*\*\*

SYSTEM-80 Level 2 16K with input level indicator, right arrow and clear keys, I/P changeover switch plus Emerson cassette recorder for external input, plus 12" B&W T.V. monitor with video I/P conversion, plus 13 months of MICRO-80 issues and cassettes, plus 10 Scotch computer cassettes

\$700.00 the lot

Mr. B. Cameron,  
15 George Street.  
PENNANT HILLS,  
N.S.W. 2120.

Tandy QUICK PRINTER II. 120 LPM Upper/Lower case on 2-3/8" aluminised paper. As new and 4 rolls of paper included

\$200.00

Dennis Bareis  
286 Lennox Street,  
MARYBOROUGH QLD. 4650  
Ph. (071) 22 1699

INTERFACE to suit System-80

\$ 40.00

TEXAS INSTRUMENTS T:59 Calculator and PC100C Thermal printer. The calculator has a card reader, 960 program steps and up to 100 memories. I have 2 extra plug in modules (each with about 22 programs) and some other software for it

\$200.00

## \*\*\*\*\* USING TRS-80 SOUND PROGRAMS ON A SYSTEM 80 - by G.F. Pukallis \*\*\*\*\*

As an owner of a System 80 machine and in possession of some system programs written for the TRS-80, I had to do something about outputting the very fancy sound routines associated with these programs.

One way of course is to alter the software and, indeed, this is what I commenced doing until I decided that I would take the easy way out and attack the exercise from the hardware angle.

Since the System 80 requires bit 2 of port "FF" to be set, (i.e. B2=1), each time output data is fed to port FF to allow the data path through relay 2 contacts, I decided that the simplest approach would be to "drive" relay 2 on permanently after loading the program.

To effect this operation a slide switch was mounted in the vicinity of the tape counter and connected into circuit as shown in FIG.1.

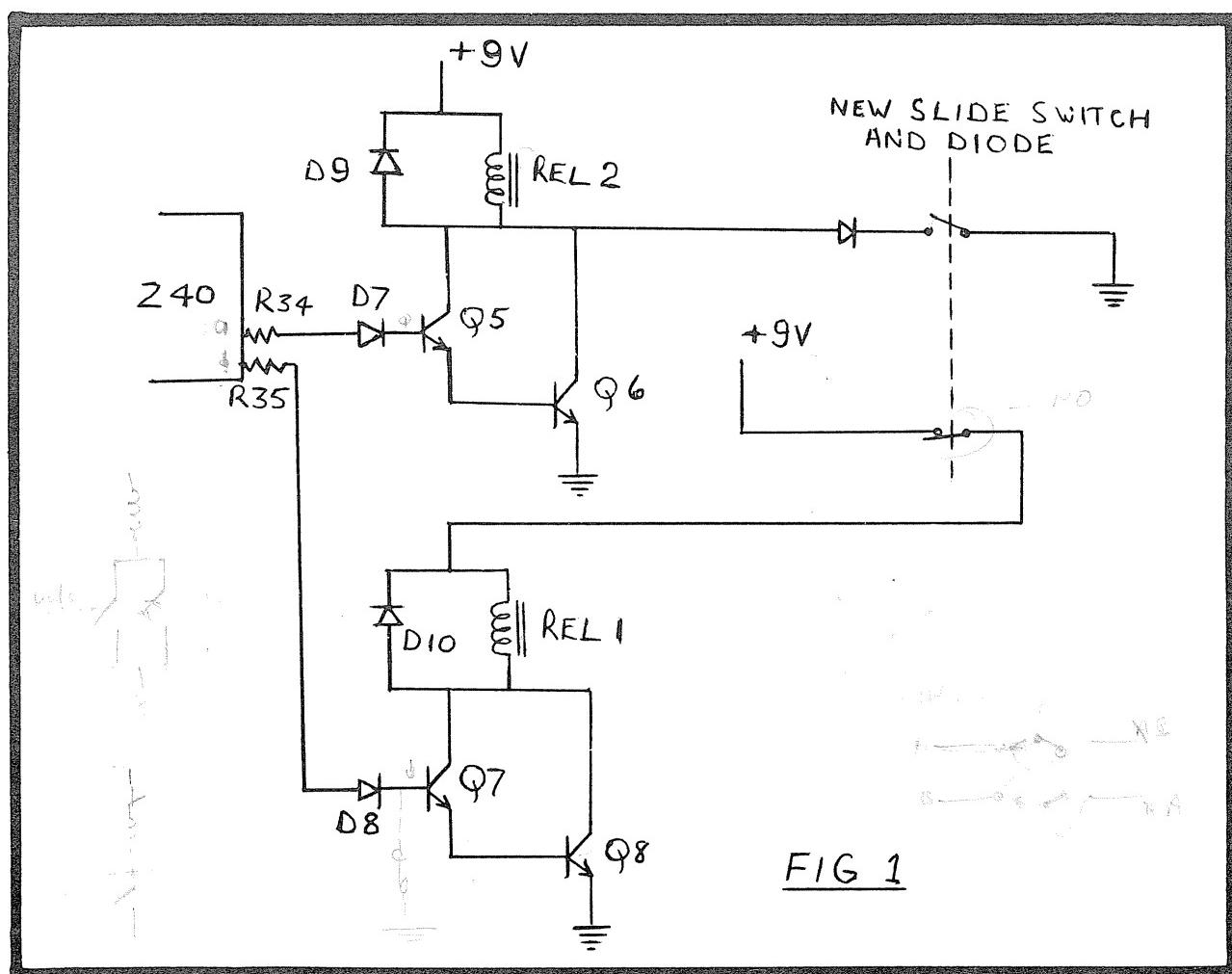
The wiring of the slide switch and diode is fairly simple, but some care has to be taken if the pencil soldering iron is too large.

The anode of the A14P diode is soldered to the anode of D9 and the hookup cable taken from the cathode to a side of a set of N/O contacts of "SWS".

The other side of this contact can be taken to ground on the emitter of Q6.

The physical layout of the interface PC board is available in the System 80 Technical Manual Issue No. 1, November 1980.

At the time of writing these notes, the second N/C contact of SWS had not yet been wired in as drawn, but I think it would be a wise move to do so as there may be some danger in both relays being energized at any one time.



As easily seen from FIG.1 the second contact of SWS simply opens the +9V rail to relay 1 when actuated.

The above alteration was overall a very simple installation and will allow full sound from the System 80 when running programs written for the TRS-80 without any software modification.

## \*\*\*\*\* READERS' REQUESTS \*\*\*\*\*

This column is a regular feature of MICRO-80. In it, we list all those articles, programs, etc. requested by our readers. We invite contributions from readers to satisfy these requests and will, of course, pay a publication fee for all articles, programs etc. printed. As a guide, we will pay a minimum publication fee of \$10 for any article or review published. In the case of software reviews, we will aim to pay in accordance with the value of the program, up to a maximum of \$25. So, if you write a good review which we publish and the usual selling price of the program in Australia is \$19.95, then we would pay you \$20. In that way, the successful reviewer will get the program he reviews, free. (Make sure you include the selling price in your review). Unfortunately, we cannot afford that policy on hardware (!) so we will pay in accordance with the merits of the review - generally of the order of \$25. Submission of a review for publication automatically means that you are prepared to accept the figure we decide to pay you and no correspondence will be entered into. Payment will be made within 30 days of publication.

## \*\* ARTICLES \*\*

- File handling on the '80
- Description of the functions performed by the Expansion Interface
- Reviews of '80 compatible printers
- Reviews of commercially available software (including that produced by us!)
- Reviews of commercially available hardware
- \* A master index to the appropriate sections in the Tandy Manuals in Level I, Level II, DOS etc.
- Comparative reviews of disk drives
- How to convert a Level I program to Level II
- A simple guide to using Level I Arrays
- \* An explanation of how to make full use of USR, PEEK and POKE statements
- Discussion of the various electric fields produced by the keyboard, tape recorder, monitor disk drives etc., how to measure them, how important they are and how to combat them

## \*\* SOFTWARE \*\*

- \* A m.l. program to enable the break key to work like RESET when using an expansion interface
- Stock market program
- \* Horse racing system
- "Files" program modified for 48K system
- \* Morse code decoder
- Sub-routine Forum
- A new STAR-TREK game
- Conversational programs (like Eliza)
- 3D programs (such as a maze seen from the inside)
- Program to scroll a Level II listing, one line at a time
- Modification to SCRIPSIT which will enable it to output to the SYSTEM 80 printer port
- Double Precision Trig, Log and Exponential machine language routines
- Programs of pharmaceutical interest such as Pharmacokinetics, Patient Medication Records, Drug information services etc.

## \*\* HARDWARE \*\*

- Interfacing the '80 to external hardware
- Review on the performance of line filters
- Real Time clock
- \* Radio Teletype/Morse interfacing
- RFI (Radio Frequency Interference) suppression
- Interface for a Teletype printer

NOTE: An \* denotes that we already have some suitable material on hand for this topic.

- 0000000000 -

## \*\*\*\*\* SOFTWARE SECTION \*\*\*\*\*

## \*\*\*\*\* BREAKOUT LI/4K (C) W. SKUZA \*\*\*\*\*

Breakout is a mad cross between tennis, squash and a demolition crew. The idea is to hit a ball towards a series of walls, knocking as many holes in them as possible. Points earned are shown under each wall. You control the bat at the left of the screen which you can move up and down. If you fail to hit a ball then the next ball is played.

You have 5 balls before the end of the game. If required, you may save your high score to tape using the inbuilt routine so that next time the game is played the previous high score can be loaded back in. The high score is displayed on the screen. If you manage to clear the board a new set of walls is drawn and play continues until the 5 balls have been used.

```

10 L=0:M=0
20 I."ENTER A NUMBER <10 - 1000> ",D
30 IF(D<10)+(D>1000)T.20
40 F.E=1TOD:H=R.(0):N.E
50 I=58:J=59:N=-23.6:O=0:P=0:Q=0:R=0:S=0
60 I."DO YOU WANT INSTRUCTIONS (1=YES, 2=NO) ";U:IFU=1G.1100
70 IFU<>2T.60
75 IFU=2G.1210
80 N=0:C.:F.Z=4T0127:S.(Z,5):S.(Z,34):N.Z
90 F.Z=5T034:S.(127,Z):N.Z
100 P.T.(21),"^> BREAKOUT <>
110 F.Z=6T033:F.A=60T0102S.6:S.(A,Z):N.A:N.Z
120 F.Z=1T08:P.A.3*Z+794,Z:N.Z
130 S.(7,43):S.(10,43)
150 S=S+1:IFS=6T.850
160 P.A.919,"BALL";S;A.833,"HIGH SCORE";O;A.881,"SCORE";P
170 F.Y=LTOM:R.(3,Y):N.Y:L=17:M=22
180 F.Y=LTOM:S.(3,Y):N.Y
190 F.Z=1TOR.(2000):N.Z:I=58:J=59:Y=R.(27)+6:K=3:N=0
200 P.A.900,:K=-K:R=0
210 R.(I,Y):R.(J,Y):IFY<=760S.750
220 IFY>=3360S.780
230 Y=Y+N:I=I+K:J=J+K:IFJ=125K=-K:R=1
240 IFI=4T.600
250 S.(I,Y):S.(J,Y):IFP.(7,43)=0G0S.500
260 IFP.(10,43)=0G0S.530
270 IFP.(J+1,Y)=0T.210
300 F.A=60T0102S.6:IFP.(J+1,Y)=P.(A,Y)T.400
310 N.A
320 G.210
360 I.#A$,B$,0
370 P."THIS HIGH SCORE OF";O;"WAS SET BY ";A$;" ON ";B$
380 F.Z=1T01500:N.Z:Q=1:RET.
400 R.(J+1,Y):P=P+(J-53)/6:IFI.(P/1008)=P/1008S=S-1:G.80
410 K=-6:P.A.881,"SCORE";P;A.900,:IFR=1K=6
420 R=0:G.210
500 IFL<3RET.
510 P.A.900,:S.(7,43):R.(3,M):R.(3,M-1):L=L-2:M=M-2
520 S.(3,L+1):S.(3,L):RET.
530 IFM>37RET.
540 P.A.900,:S.(10,43):R.(3,L+1):R.(3,L):L=L+2:M=M+2
550 S.(3,M-1):S.(3,M):RET.
600 K=-6
610 IFP.(I-1,Y)=0T.150
620 IFL=I.(Y+.5)N=-2*R.(0)
630 IFL+1=I.(Y+.5)N=-1.5*R.(0)
640 IFL+2=I.(Y+.5)N=R.(0)-.65
650 IFL+3=I.(Y+.5)N=R.(0)-.35
660 IFL+4=I.(Y+.5)N=1.5*R.(0)
670 IFL+5=I.(Y+.5)N=2*R.(0)
680 IFA.(N)<1T.200
690 IFN<ON=-1:G.200
700 IFN>ON=1:G.200
750 IFN<OT.770
760 RET.
770 Y=6:N=-N:RET.
780 IFN>OT.800
790 RET.
800 Y=34:N=-N:RET.
850 P.:P.:P.:P.:P.:IFQ=OT.950
860 IFP>OT.900
870 P.T.(25),"GOOD TRY !!!
880 P.T.(5),"BUT YOU HAVEN'T BEATEN THE PREVIOUS RECORD
890 P.T.(5),"OF";O;". BETTER LUCK NEXT TIME":G.1000
900 P.T.(25),"WELL DONE !!!
910 P.T.(8),"YOU HAVE JUST SET A NEW HIGH SCORE OF";P
920 P.T.(8)," BEATING THE PREVIOUS HIGH SCORE BY";P-O
950 Q=Q+1:O=P
1000 P=O:L=17:M=22:P.T.(13),:IN."TRY AGAIN (1=YES, 2=NO) ";U
1010 S=0:IFU=1T.80
1020 IFU=2T.1040
1030 G.1000
1040 P.:I."DO YOU WISH TO SAVE HIGH SCORE (1=YES, 2=NO) ";U
1050 IFU=2T.1250
1060 IFU<>1T.1040
1070 P."PREPARE TAPE AND SET RECORDER TO RECORD MODE

```

```

1080 I."ENTER NAME AND DATE (IE JOE BLOW,18/9/1980) ";A$,B$
1090 P.#A$;";B$;";O:G.1250
1100 C.:P.T.(21),"^> BREAKOUT <]
1110 P."THIS PROGRAM ATTEMPTS TO SIMULATE THE POPULAR ARCADE GAME.
1120 P."THE IDEA IS TO SCORE AS MANY POINTS AS POSSIBLE BY HITTING A
1130 P."BALL TOWARDS A SERIES OF WALLS AND KNOCKING THEM OUT. POINTS
1140 P."EARNED ARE SHOWN UNDERNEATH EACH WALL. YOU CONTROL A BAT, WHICH
1150 P."MOVES UP <PRESS J>, OR DOWN <PRESS SPACE BAR>. IF YOU FAIL
1160 P."TO HIT THE BALL, THE NEXT BALL IS PLAYED. YOU HAVE 5 BALLS.
1170 P."YOU MAY SAVE YOUR HIGH SCORE TO TAPE BEFORE TURNING THE '80
1180 P."OFF IF YOU WISH. THIS CAN THEN BE LOADED IN AGAIN WHEN YOU
1190 P."NEXT PLAY. HIGH SCORE AND PROGRESSIVE SCORE ARE DISPLAYED."
1195 P."IF YOU MANAGE TO CLEAR THE BOARD A NEW SET OF WALLS IS DRAWN
1197 P."AND PLAY CONTINUES UNTIL 5 BALLS HAVE BEEN COMPLETED."
1200 P.T.(15),:I."PRESS ENTER";A$
1210 I."DO YOU WISH TO LOAD DATA (1=YES, 2=NO)";U:IFU=2T.80
1220 IFU<>1T.1210
1230 P."PREPARE DATA TAPE":F.Z=1T01500:N.Z:GOS.360
1240 G.80
1250 E.

```

\*\*\*\*\* INTERCEPT LI/4K (C) M. BINET \*\*\*\*\*

This is a game of skill and cunning. The object of the game is to intercept as many targets as possible. To do this, you control a moving dotted line with the left and right arrow keys. Pressing the right arrow causes the line to change direction 90 degrees clockwise, while pressing the left arrow causes the line to change direction 90 degrees anti-clockwise. Targets in the form of white blocks will appear on the screen in a random fashion. You must intercept these targets before they disappear. If you succeed, you will receive a score between 30 and 100. However, if you run into yourself or the wall you will lose a line. You have 5 lines to start with.

```

10 REM PROGRAM BY MARLON BINET
20 REM WRITTEN SOME TIME IN AUGUST 1980
30 C.:Y=1:N=0:P."WELCOME TO INTERCEPT":IN."DO YOU WANT THE RULES";Q
40 C.:H=0:A=2:B=0:P.A.896;:IFQ=1GOS.1000
50 F.I=OT0127:S.(I,1):S.(I,2):S.(I,39):S.(I,40):N.I:F.I=3T038
60 S.(0,I):S.(1,I):S.(126,I):S.(127,I):N.I:S=0:N=5:C=0:GOS.2000
70 F.I=1T020+R.(9):X=R.(115)+6:Y=R.(30)+6:S.(X,Y):F.J=1T099:N.J:R.(X,Y)
80 N.I:S.(X,Y):GOS.3000
90 X=X+A:Y=Y+B:IF(A.(X-D-4)<5)*(A.(Y-E)<2)*(C<>0)GOS.8000
95 IFF.(X,Y)G.4000
100 S.(X,Y):IFF.(2,47)=0GOS.5000
110 IF(C=0)*(R.(10)=1)T.GOS.6000
120 IFC=1T.C=0:GOS.7000
130 IFC>1T.C=C-1
140 G.90
1000 C.:P."WELCOME TO INTERCEPT.A GAME OF SKILL AND CUNNING . ."
1010 P."THE OBJECT OF THE GAME IS TO INTERCEPT AS MANY TARGETS"
1020 P."AS POSSIBLE.TO DO THIS YOU CONTROL A MOVING DOTTED LINE"
1030 P."WITH THE LEFT AND RIGHT ARROW KEYS.PRESSING THE RIGHT ARROW"
1040 P."CAUSES THE LINE TO CHANGE COURSE 90 DEGREES CLOCKWISE."
1050 P."SIMILARLY PRESSING THE LEFT ARROW CAUSES THE LINE TO "
1060 P."CHANGE COURSE 90 DEGREES ANTI-CLOCKWISE."
1070 P."FROM TIME TO TIME TARGETS (IN THE FORM OF WHITE BARS)"
1080 P."WILL APPEAR ON THE SCREEN.YOU MUST INTERCEPT THESE TARGETS"
1090 P."BEFORE THEY DISAPPEAR.IF YOU INTERCEPT A TARGET,A"
1100 P."VALUE WILL BE ADDED TO YOUR SCORE BETWEEN -30 AND 100."
1110 P."IF HOWEVER THE LINE HITS ITSELF OR THE SURROUNDING WALL"
1120 P."IT WILL BE DESTROYED.YOU HAVE 5 LINES TO GAIN AS MANY POINTS"
1130 P."AS POSSIBLE. G O O D L U C K ! ! !"
1140 IN."TYPE ENTER TO PLAY";A$:C.:P.A.896;:RET.
2000 P.A.900;"YOUR SCORE:";S.A.919;"LINES LEFT:";N;AT943;"HIGH SCORE";H
2010 RET.
3000 P.A.961;:F.I=OT05:S.(I,47):N.I:RET.
4000 N=N-1:GOS.2000:IFN=0G.10000
4010 F.I=65T0769S.64
4020 P.A.I;"";
4030 P." ";:N.I:GOS.3000:G.70
5000 IFF.(1,47) I=B:B=A:A=-I:GOS.3000:RET.
5010 I=B:B=-A:A=I:GOS.3000:RET.
6000 F=R.(114)+1:G=R.(36)+2:F.I=FTOF+8:IFF.(I,G)C=0:RET.
6010 N.I:F.I=FTOF+8:S.(I,G):N.I:C=30+R.(10):D=F:E=G:RET.

```

**DON'T BE HELD BACK BY AN  
ANTIQUATED DISK OPERATING SYSTEM  
MOVE UP TO**

**NEWDOS 80**

NEWDOS 80 is a completely new DOS for the TRS-80 SYSTEM 80. It is well-documented, bug free and increases the power of your system many times over. It is upward compatible with TRSDOS AND NEWDOS (ie TRSDOS and NEWDOS+ programs will run on NEWDOS 80 but the reverse is not necessarily so).

These are just a few of the many new features offered by NEWDOS 80.

- \* New BASIC commands that support variable record lengths up to 4095 bytes long.
- \* Mix or match disk drives. Supports any track count from 18 to 96. Use 35, 40, 77 or 80 track 5½ inch mini disk drives, 8 inch disk drives OR ANY COMBINATION.
- \* An optional security boot-up for BASIC or machine code application programs. User never sees "DOS-READY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements, including "LIST".
- \* New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- \* Enhanced and improved RENUMBER that allows relocation of subroutines.
- \* Create powerful chain command files which will control the operation of your system.
- \* Device handling for routing to display and printer simultaneously.
- \* MINIDOS — striking the D, F and G keys simultaneously calls up a MINIDOS which allows you to perform many of the DOS commands without disturbing the resident program.
- \* Includes Superzap 3.0 which enables you to display/print/modify any byte in memory or on disk.
- \* Also includes the following utilities:
  - Disk Editor/Assembler
  - Disassembler (Z80 machine code)
  - LM offset — allows transfers of any system tape to Disk file — automatically relocated.
  - LEVEL I — Lets you convert your computer back to Level 1.
  - LVIDKSL — Saves and loads Level 1 programs to disk.
  - DIRCHECK — Tests disk directories for errors and lists them.
  - ASPOOL — An automatic spooler which routes a disk file to the printer whilst the computer continues to operate on other programs.
  - LCDVR — a lower case drives which display lower case on the screen if you have fitted a simple lower case modification.

**DISK DRIVE USERS  
ELIMINATE CRC ERRORS  
AND  
TRACK LOCKED OUT MESSAGES  
FIT A PERCOM DATA SEPARATOR  
\$37.00 plus \$1.20 p&p.**

When Tandy designed the TRS-80 expansion interface, they did not include a data separator in the disk-controller circuitry, despite the I.C. manufacturer's recommendations to do so. The result is that many disk drive owners suffer a lot of Disk I/O errors. The answer is a data separator. This unit fits inside your expansion interface. It is supplied with full instructions and is a must for the serious disk user.

**MPI DISK DRIVES  
HIGHER PERFORMANCE — LOWER PRICE**

MPI is the second largest manufacturer of disk drives in the world. MPI drives use the same form of head control as 8" drives and consequently, they have the fastest track-to-track access time available — 5msec! All MPI drives are capable of single or double-density operation. Double-density operation requires the installation of a PERCOM doubler board in the expansion interface.

As well as single head drives, MPI also makes dual-head drives. A dual-head drive is almost as versatile as two single-head drives but is much cheaper.

Our MPI drives are supplied bare or in a metal cabinet — set up to operate with your TRS-80 or SYSTEM 80. All drives are sold with a 90 day warranty and service is available through MICRO-80 PRODUCTS.

**MPI B51 40 Track Single Head Drive . . . . . only \$339  
MPI B52 40 Track Double Head Drive . . . . . only \$449**

Prices are for bare drives and include p&p. Add \$10.00 per drive for a cabinet and \$60.00 for a power supply to suit two drives. 40 track drives are entirely compatible with 35 track drives. A 40 track DOS such as NEWDOS 80 is necessary to utilise the extra 5 tracks.

**OVER 800 KILOBYTES ON ONE DISKETTE!  
WITH MPI 80 TRACK DRIVES**

MPI 80 track drives are now available. The B91 80 track single-head drive stores 204 Kilobytes of formatted data on one side of a 5½ inch diskette in single-density mode. In double-density mode it stores 408 Kilobytes and loads/saves data twice as quickly.

The B92 80 track dual-head drive stores 204 Kilobytes of formatted data on EACH side of a 5½ inch diskette in single-density mode. That's 408 Kilobytes per diskette. In double-density mode, the B92 stores a mammoth 408 Kilobytes per side or 816 Kilobytes of formatted data per diskette. With two B92's and a PERCOM double, you could have over 1.6 Megabytes of on line storage for your TRS-80 for less than \$1500!!

**MPI B91 80 Track Single Head Drive . . . . . only \$499  
MPI B92 80 Track Dual Head Drive . . . . . only \$599**

Prices are for bare drives and include p&p. Add \$10.00 per drive for a cabinet and \$60.00 for a power supply to suit two drives. Note: 80 track drives will not read diskettes written on a 35 or 40 track drive. If drives with different track counts are to be operated on the same system, NEWDOS 80 must be used.

**CARE FOR YOUR DISK DRIVES?  
THEN USE  
3M's DISK DRIVE HEAD CLEANING DISKETTES  
\$30.20 incl. p&p.**

Disk drives are expensive and so are diskettes. As with any magnetic recording device, a disk drive works better and lasts longer if the head is cleaned regularly. In the past, the problem has been, how do you clean the head without pulling the mechanism apart and running the risk of damaging delicate parts. 3M's have come to our rescue with SCOTCH BRAND, non-abrasive, head cleaning diskettes which thoroughly clean the head in seconds. The cleaning action is less abrasive than an ordinary diskette and no residue is left behind. Each kit contains:

- 2 head cleaning diskettes
- 1 bottle of cleaning fluid
- 1 bottle dispenser cap

**USE TANDY PERIPHERALS ON YOUR SYSTEM-80**

VIA

**SYSPAND-80 – \$119 incl. p&p**

The SYSTEM-80 hardware is not compatible with the TRS-80 in two important areas. The printer port is addressed differently and the expansion bus is entirely different. This means that SYSTEM-80 owners are denied the wealth of economical, high performance peripherals which have been developed for the TRS-80. Until now, that is. MICRO-80 has developed the SYSPAND-80 adaptor to overcome this problem. A completely self-contained unit in a small cabinet which matches the colour scheme of your computer, it connects to the 50-way expansion part on the rear of your SYSTEM 80 and generates the FULL Tandy 40 way bus as well as providing a Centronics parallel printer port. SYSPAND-80 enables you to run an Exatron Stringy Floppy from your SYSTEM 80, or an LNW Research expansion interface or any other desirable peripherals designed to interface to the TRS-80 expansion port. Make your SYSTEM 80 hardware compatible with the TRS-80 via SYSPAND-80.

**PROGRAMS BY MICROSOFT****EDITOR ASSEMBLER PLUS (L2/16K)****\$37.50 + \$1.20 p&p**

A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

**LEVEL III BASIC****\$59.95 plus \$1.20 p&p**

Loads on top of Level II BASIC and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries) keyboard debounce, suitable for L2/16K and up (Not Disk BASIC)

**ADVENTURE ON DISK****\$35.95 plus \$1.20 p&p**

This is the original ADVENTURE game adapted for the TRS-80. The game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K One Disk)

**BASIC COMPILER****\$208 plus \$2.00 p&p**

New improved version, the Basic Compiler converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate.

**UPGRADE TO 16K  
FOR ONLY \$30.00!!****MICRO-80's 16K MEMORY EXPANSION KIT  
HAS BEEN REDUCED IN PRICE EVEN MORE**

Larger volume means we buy better and we pass the savings on to you. These are our proven, prime, branded 200 ns (yes, 200 nanosecond) chips. You will pay much more elsewhere for slow, 350 ns. chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory in the TRS-80 — these cost an additional \$4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.

**DISK DRIVE CABLES  
SUITABLE FOR ANY DISK DRIVES**

DC-2 2 Drive Connector Cable . . . . . \$35 incl. p&amp;p

DC-4 4 Drive Connector Cable . . . . . \$45 incl. p&amp;p

**DOUBLE THE SPEED AND CAPACITY****OF YOUR DISK DRIVES****PERCOM DOUBLER ONLY \$220****plus \$2.00 p&p**

Installing a Doubler is like buying another set of disk drives, only much cheaper!! The doubler works with most modern disk drives including:- MPI, Micropolis, Pertec, TEAC (as supplied by Tandy). The doubler installs in the TRS-80 expansion interface, the System-80 expansion interface and the LNW Research expansion interface in a few minutes without any soldering, cutting of tracks, etc. It comes complete with its own TRSDOS compatible double density operating system.

**DOUBLE-ZAP II – DOUBLE DENSITY PATCH****FOR NEWDOS 80****ONLY \$53.00 plus \$1.00 p&p**

If you are using NEWDOS 80, then you also need DOUBLE-ZAP II, on diskette. This program upgrades your NEWDOS 80 to double density with ADR (automatic density recognition.) It retains all the familiar features, including the ability to mix and match track counts on the same cable. In addition, it gives NEWDOS 80 the ability to mix densities on the same cable, automatically. If you place a single density diskette in drive 0, say and a double density diskette in drive 1, Double-ZapII will recognise this and read/write to drive 0 in single density whilst at the same time it reads/writes to drive 1 in double density!

**FLOPPY DOCTOR AND MEMORY DIAGNOSTIC  
(by MICRO CLINIC) \$29.95 plus 50c. p&p**

Two machine language programs on a diskette together with manual which thoroughly test your disk drives and memory. There are 19 possible error messages in the disk drive test and their likely causes are explained in the manual. Each pass of the memory tests checks every address in RAM 520 times, including the space normally occupied by the diagnostic program itself. When an error occurs the address, expected data, and actual data are printed out together with a detailed error analysis showing the failing bit or bits, the corresponding IC's and their location. This is the most thorough test routine available for TRS-80 disk users.

**BOOKS****LEVEL II ROM REFERENCE MANUAL****\$24.95 + \$1.20 p&p**

Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

**TRS-80 DISK AND OTHER MYSTERIES****\$24.95 + \$1.20 p&p**

The hottest selling TRS-80 book in the U.S.A. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories — this is a must for the serious Disk user and is a perfect companion to any of the NEWDOS's.

**LEARNING LEVEL II****\$16.95 + \$1.20 p&p**

Written by Daniel Lien, the author of the TRS-80 Level I Handbook, this book teaches you, step-by-step, how to get the most from your Level II machine. Invaluable supplement to either the TRS-80 Level II Manual or the System-80 Manuals.

**HIGH QUALITY DISKETTES** ALL PRICES INCLUDE P&P

40 TRACK NASHUA SINGLE SIDE/SINGLE DENSITY ... \$45.00 box of 10

**HIGH QUAL**

40 TRACK VERBATIM DOUBLE SIDE

# MORE AUSTRALIAN SOFTWARE

All programs designed to run on both the TRS-80 or the SYSTEM 80 without modification. Most programs include sound

## TRIAD VOL 1 – L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

Three separate games which test your powers of memory and concentration. The programs combine graphic displays and sound:

**SIMON-SEZ:** Just like the electronic music puzzles on sale for more than \$20. Numbers are flashed on the screen and sounded in a sequence determined by the computer. Your task is to reproduce the sequence, correctly.

**LINE?**: Rather like a super, complicated version of noughts and crosses. You may play against another player or against the computer itself. But beware, the computer cheats!

**SUPER CONCENTRATION:** Just like the card game but with more options. You must find the hidden pairs. You may play against other people, play against the computer, play on your own, or even let the '80 play on its own.

## TRIAD VOL 2 – L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

Remember those "NUMERO" puzzles in which you had a matrix of numbers (or letters) with one blank space and you had to shuffle the numbers around one at a time until you had made a particular pattern? Well, **SHUFFLEBOARD**, the first program in this triad, is just this, except that the computer counts the number of moves you take to match the pattern it has generated — so it is not possible to cheat.

**MIMIC** is just like SHUFFLEBOARD except that you only see the computer's pattern for a brief span at the beginning of the game, then you must remember it!

In **MATCHEM**, you have to manoeuvre 20 pegs from the centre of the screen to their respective holes in the top or bottom rows. Your score is determined by the time taken to select a peg, the route taken from the centre of the screen to the hole and your ability to direct the peg into the hole without hitting any other peg or the boundary.

## VISURAMA L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

Two programs which give fascinating, ever-changing patterns on the screen.

**LIFE** is the fastest implementation of the Game of Life you will see on your '80. Machine language routines create up to 1200 new generations per minute for small patterns or up to 100 per minute for the full 128 x 48 screen matrix. Features full horizontal and vertical wraparound.

**EPICYCLES** will fascinate you for hours. The ever-changing ever-moving patterns give a 3D effect and were inspired by the ancient Greek theories of Ptolemy and his model of the Solar system.

## EDUCATION AND FUN – L1/4K, L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

Written by a primary school teacher to make learning enjoyable for his pupils, there are five programs in both Level I and Level II to suit all systems:

**BUG-A-LUG:** a mathematics game, in which you must get the sum correct before you can move.

**AUSTRALIAN GEOGRAPHY:** learn about Australian States and towns, etc.

**SUBTRACTION GAME:** build a tower with correct answers.

**HOW GOOD IS YOUR MATHS?** Select the function (+, -, ÷ or X) and degree of difficulty.

**HANGMAN:** That well known word game now on your computer.

*Recommended for children from 6 to 9 years.*

## \*\*\*SAVE 00\$'s \*\*\*SAVE 00\$'s \*\*\*SAVE 00\$'s \*\*\*MICRO-80 EXPANSION INTERFACE \*\*\*

MICRO-80's expansion interface utilises the proven LNW Research Expansion board. It is supplied fully built up and tested in an attractive cabinet with a self contained power supply, ready to plug in and go. The expansion interface carries MICRO-80's full, no hassle, 90-day warranty.

Features include:- • Sockets for up to 32K of memory expansion • Disk controller for up to 4 disk drives • Parallel printer port • Serial RS232C/20mA I/O port • Second cassette (optional)

The expansion interface connects directly to your TRS-80 L2/16K keyboard or, via SYSPAND-80 to your SYSTEM-80VIDEO GENIE Prices: HD-010-A Expansion Interfaces with Ø K : \$415.00 HD-010-B Expansion Interfaces with 32K : \$475.00 HD-011 Data separator fitted (recommended) : add \$25.00 HD-012 Dual cassette Interfaces fitted : add \$15.00.

The MICRO-80 Expansion Interface is also available in kit form.

Prices: HD-013 Kit consisting of LNW Research PC board and manual, ALL components including cabinet & power supply : \$335.00 HD-011 Data separator for above : \$22.00 HD-013 Dual cassette Interface kit : \$12.00.

## COSMIC FIGHTER & SPACE JUNK – L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

Both programs have sound to complement their excellent graphics. In **COSMIC FIGHTER**, you must defend the earth against seven different types of alien aircraft. It is unlikely that you will be successful but you will have a lot of fun trying!

Your mission in **SPACE JUNK** is to clean up all the debris left floating around in space by those other space games. It is not as simple as it sounds and space junk can be quite dangerous unless you are very careful.

## SPACE DRIVE L2/4K & 16K

Cassette \$7.95 Disk \$12.95

+ 60c p&p

Try to manoeuvre your space ship through the meteor storms then land it carefully at the space port without running out of fuel or crashing. Complete with realistic graphics.

## STARFIRE AND NOVA INVASION L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

Both programs include sound to improve their realism.

**STARFIRE** seats you in the cockpit of an X-wing fighter as you engage in battle with the deadly Darth Vader's Tie-fighters. Beware of the evil one himself and may the Force be with you.

In **NOVA INVASION**, you must protect your home planet of Hiberna from the invading NOVADIANS. You have two fixed guns at each side of the screen and a moveable one at the bottom. Apart from shooting down as many invaders as possible, you must protect your precious hoard of Vitamium or perish!

## AIR ATTACK AND NAG RACE – L2/16K

Cassette \$9.95 Disk \$14.95

+ 60c p&p

An unlikely combination of programs but they share the same author who has a keen sense of humour.

**AIR ATTACK** includes sound and realistic graphics. The aircraft even have rotating propellers! But they also drop bombs on you, so it's kill or be killed!

**NAG RACE** lets you pander to your gambling instinct without actually losing real money. Up to five punters can join in the fun. Each race results in a photo-finish whilst there is a visible race commentary at the bottom of the screen throughout the race. Happy punting!

## FOUR LETTER MASTERMIND L2/16K

Cassette \$7.95 Disk \$12.95

+ 60c p&p

There are 550 four-letter words from which the computer can make its choice. You have 12 chances to enter the correct word. After each try, the computer informs you of the number of correct letters and those in the correct position. You can peek at the list of possible words but it will cost you points. Makes learning to spell fun.

## MUSIC IV – L2/16K

Cassette \$7.95 Disk \$12.95

+ 60c p&p

Music IV is a music compiler for your '80. It allows you to compose or reproduce music with your computer that will surprise you with its range and quality. You have control over duration (full beat to 1/16 beat) with modifications to extend the duration by half or one third for triplets. Both sharps and flats are catered for as are rests. Notes on whole sections may be repeated. The program comes with sample data for a well-known tune to illustrate how it is done.

**TURN  
THIS**

**into  
this**

**for \$49.00**

**plus \$2.00 p & p**

A choice of upper and lower case display is easier to read, gives greater versatility.

The Micro-80 lower case modification gives you this facility, plus the symbols for the 4 playing-card suits for \$49.00 + \$2.00 p & p.

The Micro-80 modification features true below-the-line descenders and a block cursor.

Each kit comes with comprehensive fitting instructions and two universal lower-case drive routines on cassette to enable you to display lower case in BASIC programs.

The driver routines are self-relocating, self-protecting and will co-reside with other machine language programs such as Keyboard-debounce, serial interface driver programs etc.

Both programs give your TRS-80™ Model I or System 80™ an optional typewriter capability, i.e. shift for upper case.

The second programme also includes Keyboard-debounce and a flashing cursor.

You fit it. Or we can.

Fitting the modification requires soldering inside the computer. This should only be carried out by an experienced hobbyist or technician.

If you are at all dubious, a fitting service is available in all capital cities for only \$20.00.

A list of installers is included with each kit.

# **Save \$120 now.**

**ADD A DISK DRIVE TO YOUR TRS-80™ MODEL III  
FOR ONLY \$875.00 OR ADD TWO FOR ONLY \$1199.**



The Micro-80 disk drive upgrade for the TRS-80™ Model III contains the following high quality components:

1 or 2 MPI 40-track single head disk drives, 1 VR Data double-density disk controller board and 1 dual drive power supply plus all the necessary mounting hardware, cables and comprehensive fitting instructions, which can be carried out with a minimum of fuss by any average computer owner.

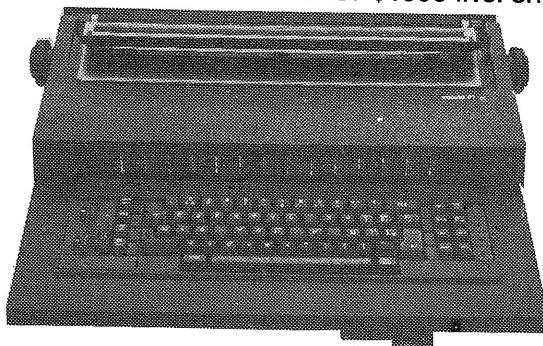
Fitting service is available for \$25.00 in most capital cities.

**ONLY \$1995 INC. S.T.**

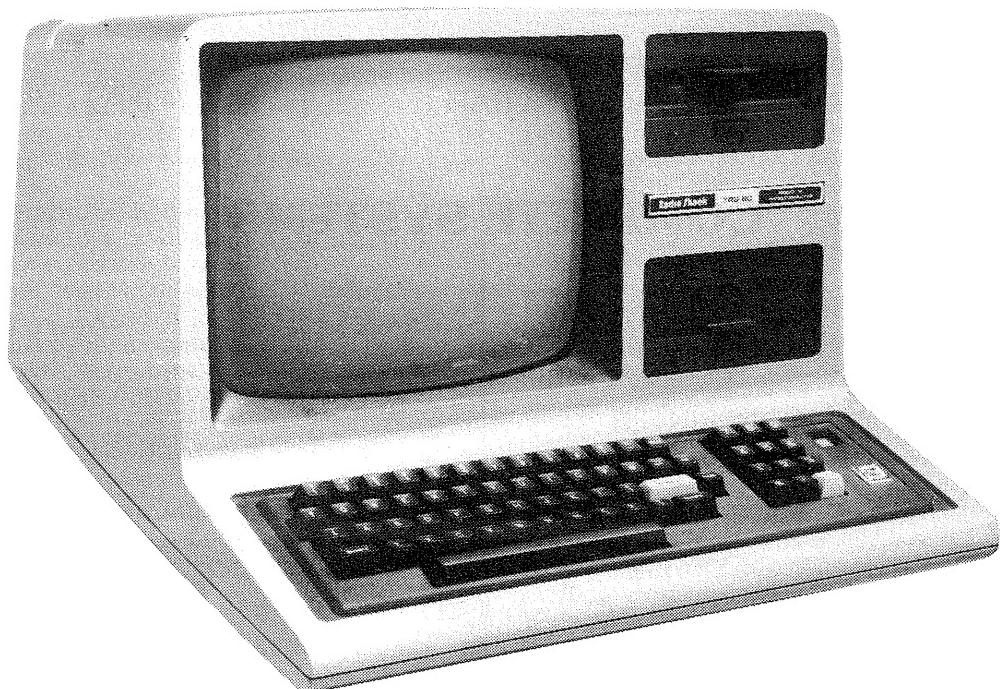
## **Daisy Wheel Typewriter/Printer**

MICRO-80 has converted the new OLIVETTI ET-121 DAISY WHEEL typewriter to work with the TRS-80 and SYSTEM 80 or any other microcomputer with a Centronics parallel port (RS 232 serial interface available shortly). The ET-121 typewriter is renowned for its high quality, fast speed (17 c.p.s.), quietness and reliability. MICRO-80 is renowned for its knowledge of the TRS-80/SYSTEM 80 and its sensible pricing policy. Together, we have produced a dual-purpose machine:- an attractive, modern, correcting typewriter which doubles as a correspondence quality Daisy-wheel printer when used with your micro-computer.

How good is it? - This part of our advertisement was typeset using an ET-121 driven by a TRS-80. Write and ask for full details.



**1.4 MEGABYTES ON LINE + 48K RAM  
for \$3750 incl. Sales Tax**



**MICRO-80's  
MODEL 380 +**

**MICRO-80** has equipped the TRS-80 with two high reliability dual-head 80 track mini-floppy disk drives made by MPI, one of America's leading mini-disk drive manufacturers.

This turns the mild-mannered Model 3 into a powerhouse able to handle the most difficult business programs. The TRS-80 is one of the best-supported microcomputers in the world. MICRO-80 has been supporting the TRS-80 in Australia for 18 months and is one of Australia's leading dealers in MPI disk drives.

**2.8 MEGABYTES FOR \$5250 incl. Sales Tax**

If you need even more file space you can add MICRO-80's external dual-drive cabinet enclosing two more dual-head 80 track drives for an additional \$1500.

# COMPUTER PRICES

## MODEL 340

2 40 TRACK SINGLE HEAD DRIVES GIVING  
350K FORMATTED STORAGE, 48K RAM

\$2850 INCL. SALES TAX

## MODEL 340 +

2 40 TRACK DUAL-HEAD DRIVES GIVING  
700K FORMATTED STORAGE, 48K RAM

\$3250 INCL. SALES TAX

## MODEL 380

2 80 TRACK SINGLE HEAD DRIVES GIVING  
700K FORMATTED STORAGE, 48K RAM

\$3250 INCL. SALES TAX

## MODEL 380 +

2 80 TRACK DUAL-HEAD DRIVES GIVING  
1.4 MEGABYTE FORMATTED STORAGE, 48K RAM

\$3750 INCL. SALES TAX

## COMPLETE SYSTEMS

### 350K SYSTEM

MODEL 340, EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$3900 INCL. SALES TAX

### 700K SYSTEM (40 Track)

MODEL 340 +, EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4300 INCL. SALES TAX

### 700K SYSTEM (80 Track)

MODEL 380, EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4300 INCL. SALES TAX

### 1.4 MEGABYTE SYSTEM

MODEL 380 +, EPSON MX-80 PRINTER  
DOSPLUS DISK OPERATING SYSTEM

\$4800 INCL. SALES TAX

### 2.8 MEGABYTE SYSTEM

MODEL 380 +, DUAL EXTERNAL DRIVES,  
MX-80 PRINTER, DOSPLUS DISK OPERATING SYSTEM

\$6300 INCL. SALES TAX

## 40 TRACK COMPATIBILITY

**MAKE 80** — PROGRAM TO CONVERT  
40 TRACK DISKS TO 80 TRACK

\$19.95

**COMPAT-80** — HARDWARE DEVICE TO SWITCH  
80 TRACK DRIVES TO 40 TRACK. GIVES FULL  
BI-DIRECTIONAL COMPATIBILITY  
(AVAILABLE SEPTEMBER '81)

\$65.00

All prices subject to change without notice. Prices are F.O.B. Adelaide. All computers and systems carry MICRO-80's  
90-day Warranty covering parts and labour.



## HIGH QUALITY DISKETTES

Despite the increases in duty and Sales Tax, MICRO-80 is able to offer high quality  
5 1/4 inch diskettes at bargain prices . . .

40 Track Scotch Brand Single Side / Single Density .....	\$55 box of 10
40 Track Verbatim Double Side / Double Density .....	\$59 box of 10
77 Track Verbatim Single Side / Single Density .....	\$59 box of 10



# SOFTWARE BY AUSTRALIAN AUTHORS

All our software is suitable for either the SYSTEM 80 or the TRS-80

## NEW SOFTWARE FROM MICRO-80 PRODUCTS BUSINESS PROGRAMS

### MICROMANAGEMENT

#### STOCK RECORDING SYSTEM (L2/16K)

Cassette version . . . . . \$29.95 + \$1.00 p&p  
Stringy Floppy version . . . . . \$33.95 + \$1.00 p&p

This system has been in use for 9 months in a number of small retail businesses in Adelaide. It is therefore thoroughly debugged and has been tailor made to suit the requirements of a small business. MICROMANAGEMENT SRC enables you to monitor the current stock level and reorder levels of 500 different stock items per tape or wafer. It includes the following features:—

- Add new items to inventory
- Delete discontinued items from inventory
- List complete file
- Search for any stock number
- Save data to cassette or wafer
- Load data from cassette or wafer
- Adjusts stock levels from sales results and receipt of goods
- List all items requiring reordering

We can thoroughly recommend this program for the small business with a L2/16K computer.

### SCOTCH BRAND COMPUTING CASSETTES

Super-quality personal computing cassettes.

C-10 pack of 10 . . . . . \$26.00 incl. p&p  
C-30 pack of 10 . . . . . \$28.00 incl. p&p

## UTILITIES

### S-KEY by Edwin Paay \$15.95 plus 50c. p&p

S-KEY is a complete keyboard driver routine for the TRS-80 and becomes part of the Level II basic interpreter. With S-KEY loaded the user will have many new features not available with the standard machine.

#### S-KEY features:

- \* S-KEY provides an auto-repeat for all the keys on the keyboard. If any key is held down longer than about half a second, the key will repeat until it is released.
- \* Graphic symbols can be typed direct from the keyboard, this includes all 64 graphic symbols available from the TRS-80/SYSTEM 80.
- \* S-KEY allows text, BASIC commands and/or graphics to be defined to shifted keys. This makes programming much easier as whole commands and statements can be recalled by typing shift and a letter key.
- \* Because S-KEY allows graphics to be typed directly from the keyboard, animation and fast graphics are easily implemented by typing the appropriate graphics symbols directly into PRINT statements.
- \* S-KEY allows the user to LIST a program with PRINT statements containing graphics, properly. S-KEY does this by intercepting the LIST routine when necessary.
- \* S-KEY allows the user to list an updated list of the shift key entries to the video display or line printer.
- \* S-KEY can be disabled and enabled when required. This allows other routines which take control of the keyboard to run with S-KEY as well.

Each cassette has TRS-80, DISK and SYSTEM 80 versions and comes with comprehensive documentation.

### BMON by Edwin Paay \$19.95 plus 50c. p&p THE ULTIMATE HIGH MEMORY BASIC MONITOR L2/16-48K

Our own personnel refuse to write BASIC without first loading this amazing machine language utility program into high memory! BMON Renumbers; Displays BASIC programs on the screen while they are still loading; tells you the memory locations of the program just loaded; lets you stop a load part-way through; merges two programs, with automatic renumbering of the second so as to prevent any clashes of line numbers; recovers your program even though you did type NEW: makes one program invisible while you work on a second (saves hours of cassette time!); lists all the variables used in the program; makes SYSTEM tapes; lets you Edit memory directly . . . the list goes on and on. Cassette comes with 16K, 32K and 48K versions, ready to load. Can anyone afford NOT to have BMON?

## EDUCATIONAL

### RPN CALCULATOR (L2/16K & 32K)

\$14.95 \$ 50c. p&p

Give your computer the power of a \$650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32K programmable version to run in 16K. Whether for business or pleasure, this package will prove invaluable, and turn you '80 into a very powerful instrument.

## GAMES

### MICROPOLY (L2/16K) \$7.50 \$ 50c p&p

Now you can play Monopoly on your micro. The old favourite board game has moved into the electronic era. This computer version displays the board on the screen, obeys all the rules and, best of all, the banker does not make mistakes with your change!

### CONCENTRATION (L2/16K) \$7.50 + 50c p&p

Another application of supergraphics. There are 28 "cards" displayed on the screen, face down. Players take it in turn to turn them over with the object of finding matching pairs. There are 40 different patterns which are chosen at random, so the game is full of endless variety. This is of particular value in helping young children to learn the art of concentrating and, at the same time, to introduce them to the computer.

### METEOR AND TORPEDO ALLEY (L2/16K)

\$9.95 + 50c p&p

Those who frequent games arcades will recognize these two electronic games. In METEOR you must destroy the enemy space ships before they see you. In its most difficult mode, the odds are a thumping 238 to 1 against you being successful. In torpedo alley you must sink the enemy ships without hitting your own supply ship. Both games include sound effects and are remarkably accurate reproductions of the arcade games.

## DISKETTES

DOUBLE DENSITY ... \$59.00 box of 10

77 TRACK VERBATIM SINGLE SIDE/SINGLE DENSITY ... \$59.00 box of 10

## HIGH QUALITY DISKETTES

**AUSTRALIAN SOFTWARE (Cont.)****GAMES****SHEEPDOG (L2/16K)****\$7.50 + 50c p&p**

Ever wondered how a sheepdog manages to drive all those awkward sheep into a pen? Well, here is your chance to find out just how difficult it is and have a lot of fun at the same time. You control the sheepdog, the computer controls the sheep! As if that isn't enough, look out for the dingoes lurking in the bush!

**U BOAT****\$7.50 plus 50c p&p**

Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedoes, plenty of targets, working fuel and battery meters, helpful Mothership for high-seas reprovisioning and even has emergency radio for that terrible moment when the depth charges put your crew at risk. Requires Level II/16K.

**SPACE INVADERS WITH SOUND****\$7.50 + 50c p&p**

Much improved version of this arcade favourite with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II with 4K and 16K versions on this cassette.

**GOLF (L2/16K)****\$7.50 + 50c p&p**

Pit your skills of mini-golf against the computer. Choose the level of difficulty, the number of holes and whether you want to play straight mini golf or crazy golf. Complete with hazards, water traps, bunkers and trees. Great fun for kids of all ages.

**DOMINOES(L2/16K)****\$7.50 + 50c p&p**

Pit your skill at dominoes against the computer, which provides a tireless opponent. Another application of supergraphics from the stable of Charlie Bartlett. Dominoes are shown approximately life size in full detail (except for colour!). The monitor screen is a window which you can move from one end of the string of dominoes to the other. Best of all, you don't lose any pieces between games!

**KID'S STUFF (formerly MMM-1) \$7.50 plus 50c. p&p**

Three games on one cassette from that master of TRS-80 graphics, Charlie Bartlett. Includes INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNIEVEL (as in motorcycle, ramp and buses).

**OTHER PROGRAMS****INFINITE BASIC BY RACET (32K/1 DISK)****\$49.95 + 50c. p&p**

Full matrix functions — 30 BASIC commands; 50 more STRING functions as BASIC commands.

**GSF/L2/48K****\$24.95 + 50c. p&p**

18 machine language routines including RACET sorts.

**BUSINESS ADDRESS AND INFORMATION SYSTEM (48K/DISK)****\$24.95 + 50c. p&p**

Allows you to store addresses and information about businesses, edit them and print them out.

**HISPED (L216, 32 or 48K) \$29.95**

This machine language program allows you to SAVE and LOAD programs and data to tape at speeds up to 2000 baud (4 times normal) using a standard cassette recorder. A switch must be installed to remove the XRX III loading board, if fitted.

**LOWER CASE FOR YOUR TRS-80/SYSTEM 80****Kit only \$49.00 plus \$2.00 p&p**

Give your TRS-80 or SYSTEM 80 a lower case display with proper descenders and a block cursor (similar to the TRS-80 Model III). Also includes symbols for the four suits of cards. Includes full fitting instructions, all necessary components and a special machine language driver program to enable lower case in BASIC. The modification is similar to the Tandy model and does not work with Electric Pencil without further modifications.

These kits require disassembly of your computer and some soldering. They should only be installed by someone who has experience in soldering integrated circuits, using a low power, properly earthed soldering iron. If you do not have the necessary experience/equipment, we will install the modification for you for \$20 plus freight in both directions. Make sure you arrange the installation with us first, before despatching your computer, so that we can assure you of a rapid turn-around. We are also arranging to have installers in each State. See elsewhere in this issue for their names and addresses.

**PRICES**

Cat No.

HD-020 Lower case mod kit for TRS-80

\$49.00 plus \$2.00 p&amp;p

HD-021 Lower case mod kit for SYSTEM-80

\$49.00 plus \$2.00 p&amp;p

**EPSON MX-80 PRINTER****ONLY \*\$949 Inc. Cable for TRS-80 and p&p**

(\*Printer only — \$940 incl. p&amp;p)

The EPSON MX-80 printer is compact, quiet, has features unheard of only 2-3 years ago in a printer at any price and, above all, is ultra-reliable. All available print modes may be selected under software control. Features include:

- high quality 9x9 dot-matrix character formation
- 3 character densities
  - . 80 characters per line at 10 chars/inch
  - . 132 characters per line at 16.5 chars/inch
  - . 40 characters per line at 5 chars/inch
- 2 line spacings
  - . 6 lines per inch . 8 lines per inch
- 80 characters per second print speed
- bi-directional printing
- logical seeking of shortest path for printing
- lower case with descenders
- TRS-80 graphics characters built in
- standard Centronics printer port

The bi-directional printing coupled with the logical seeking of the shortest print path (which means that the print head will commence printing the next line from the end which requires the least travel, thereby minimising unutilised time) gives this printer a much higher throughput rate than many other printers quoting print speeds of 120 c.p.s. or even higher.

**GREEN SCREEN SIMULATOR****\$19.95 incl. p&p**

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image.

All editorial staff of MICRO-80 are now using GREEN SCREEN SIMULATORS on their own monitors.

Please make sure to specify whether you have an old (squarish) or new (rounded) style monitor when ordering. Not available for Dick Smith monitors.

```

7000 F.I=FTOF+8:R.(I,G):N.I:RET.
8000 Z=R.(13)*10-40:F.I=OT0ZS.(Z/A.(Z))*10:S=S+(Z/A.(Z))*10:F.J=1T0350
8010 N.J=GOS.2000:IFZ<>ON.I
8020 GOS.3000:GOS.7000:C=O:RET.
10000 C.:P." *** GAME OVER ***":IFS>HP."NEW HIGH SCORE ! ! !"
10010 P."YOUR SCORE:";S,:P."","HIGH SCORE:";H:IFS>HH=S
10020 N=0:Y=1:IN."CARE TO PLAY AGAIN";Q:IFQ=1C.:P.A.896,:G.50

```

[REDACTED]

\*\*\*\*\* PRIME GENERATOR m/l (C) B. HOFF \*\*\*\*\*

What is a prime? An ancient counting system that went out of fashion in 2000 B.C? No, a prime is a number which is exactly divisible (i.e. no remainder) by only two numbers - itself and 1. 7 is prime, as it can only be divided (from now on assume "exactly divisible") by 7 and 1, but 8 is not prime, for its divisors are 1, 2, 4 and 8. How do you generate primes? I have rusty memories of small programs that asked for a number, then performed a for next loop trying out all the integers up to the square root of the number, testing for exact divisibility, i.e.:

```

10 INPUT"TYPE IN A NUMBER";A
20 FOR I% = 2 TO SQR(A)
30 IF A/I% = INT(A/I%)THEN 70
40 NEXT I%
50 PRINT A;" IS A PRIME."
60 STOP
70 PRINT A;" IS NOT PRIME."
80 END

```

But this sort of routine had its problems, divisions weren't fast, so the computer seemed struck dumb on any number above about 500000 and the working wasn't terribly efficient. Why test for 8 after 2 has been ruled out or 9 after 3 has been tried, for any number that is not divisible by 3 is certainly not divisible by 9, and what about getting lists or tables of primes. Won't the division - loop method be painfully slow.

The next addition to the program was based on the fact that "A number is prime if and only if it cannot be exactly divided by all previous primes" (up to the number's square root). This was an improvement because there was no double testing of factors and the for next loop had less work to do. This program used an array, however, and especially on 16K machines the memory limit was easily reached. Divisions still slowed things down and efficiency could not be improved noticeably until a better way of finding primes could be found. Then one day a very neat little way of computing lists of primes fell into the author's possession. Its method of operation is best described by example: Suppose we want to generate all the primes up to 10. First, we set up a list of all the integers from 2 to 10, i.e.

2 3 4 5 6 7 8 9 10

Now, we come along to the first number in the list, 2 and we cross off all the multiples of 2, excluding 2 itself. The list then reads:

2 3 5 7 9

The next number that has not been crossed off is 3, so again all multiples are crossed off, leaving:

2 3 5 7

The next uncrossed number in the list is 5, but since this is beyond the square root of the limit of primes we are doing, no further calculations are needed. The last thing that needs to be done is that all the numbers not crossed out are printed, i.e. 2 3 5 7 is the result of this run. Note that only additions need be used - no division or multiplication is needed. Note also that the routine gets faster as it goes, the first loop being the slowest, then successive loops having to cross less and less numbers.

\*\*\*\*\*

ED. That great splodge of asterisks is to make sure you notice this line and the fact that the listing supplied below by the author is NOT Microsoft BASIC and therefore of course needs to be changed before it would run in an '80 and, anyway, it is only supplied as an example and why bother converting the listing anyway as his m/l program does the same thing, only faster and that DOES run on an '80. Another great splodge of asterisks coming up (I had some left over from the last time).

\*\*\*\*\*

A BASIC program to do the generation of primes as given above went as follows:

```

1 OPTION SPREC      define No. handling as single precision
10 DIMA$(1)32767) string array, 32767 elements, length 1 each
20 I=3             we will generate only the odd Nos in list
30 A$(I)="A"
40 I=I+2
50 IF I<32767 THEN 30 ELSE I=3
70 IF A$(I)=""THEN 120 ELSE J=I+I
90 A$(J)=""        cross off a number from the list
100 J=J+I
110 IF J<32768 THEN 90 have we finished crossing off numbers
120 I=I+2          next number in the list
130 IF I<182 THEN 70 182 is the square root of 32767
140 SKIP           go to a new page (ed. CLS maybe)
150 PRINT" 2":    2 is not generated so it must be printed
160 I=3            for printing out nos not crossed off
170 IF A$(I)="A" THEN PRINT USING 220,I
180 I=I+2          look for next number to print out
190 IF I<32768 THEN 170
200 PRINT
210 SKIP
220 FORM PIC(zzzzz#) microsoft equivalent ?USEING"#####";I;
in line 170
230 END

```

Despite being in IBM V/S BASIC, most of the code above is straightforward. I is a pointer indicating the position in the list of the next number to be crossed off. J is the pointer used for crossing off numbers. Only additions and compares are used, making the routine easy to transcribe into machine language. Note also that the FOR NEXT loops are converted into increment and compares, originally to help attain maximum speed but also making conversion into machine code easy and painless.

Finally then we get to the program that this article is all about. The IBM listing above took 1.47 minutes to run primes up to 32767 compared to this machine language version which will run primes up to 65535 in 14 seconds.

#### HOW TO LOAD

Using a suitable monitor enter the machine code into your system and save it to tape or enter the assembly listing into your machine using EDTASM.

	LOAD ADDRESS		
	START	END	ENTRY
HEX.	4300	43D5	4300
DEC.	17152	17365	17152

To run the program answer MEMORY SIZE? by pressing the ENTER/NEWLINE key then type SYSTEM and answer the response with the program name PRYME and press the ENTER/NEWLINE key.

Answer the second response \*? with:- /17152 and press the ENTER/NEWLINE key. The screen will go blank for a few seconds and then the prime numbers will be listed on the screen in columns. At the completion of the program it will return you to BASIC with READY

The source code listing has been written for the MICROSOFT Editor/Assembler Plus and will not assemble correctly via the Radio Shack Editor Assembler as macros are used. Those who do not have the Editor/Assembler Plus may enter the object code shown in the two left-hand columns using a suitable monitor program such as BMON.

		ORG	4300H
4300	00100	MACRO	#A
	00110 BITS	PUSH	HL
	00120	PUSH	BC
	00130	XOR	A
	00140	RR	H
	00150	RR	L
	00160	JR	NC, P#\$YM
	00170	LD	B, 3
	00180	XOR	A
	00190 A##\$YM	RR	H
	00200	RR	L
	00210	RRA	
	00220	DJNZ	A##\$YM
	00230	RLCA	
	00240	RLCA	
	00250	RLCA	
	00260	LD	B, A
	00270	INC	B
	00280	XOR	A
	00290	SCF	
	00300	RLA	
	00310 B##\$YM	DJNZ	B##\$YM
	00320	LD	BC, BASE
	00330	ADD	HL, BC
	00340	COND	#A=2
	00350	CPL	
	00360	ENDC	
	00370	AND	(HL)
	00380	COND	#A=2
	00390	LD	(HL), A
	00400	ENDC	
	00410	POP	BC
	00420 P##\$YM	POP	HL
	00430	ENDM	
4300	21A643	START	LD HL, MESS
4303	CDA728	00450	CALL 28A7H
4306	CD4900	00460	CALL 49H
4309	FE59	00470	CP 'Y'
430B	2005	00480	JR NZ, GO
430D	3E01	00490	LD A, 1
430F	329C40	00500	LD (409CH), A
4312	0612	00510	LD B, 12H
4314	3E0D	00520	LD A, 0DH
4316	CD3A03	00530	LD GO2
4319	10FB	00540	CALL 33AH
431B	21D443	00550	DJNZ GO2
431E	36FF	00560	LD HL, BASE
4320	11D543	00570	LD (HL), 0FFH
4323	010110	00580	LD DE, BASE+1
		00590	LD BC, 1001H
4326	EDB0	00600	LD IR
4328	210300	00610	LD HL, 3
432B	CDAF43	00620	CALL TEST
432E	44	00630	LD B, H
432F	4D	00640	LD C, L
4330	282F	00650	JR Z, L120
4332	09	00660	ADD HL, BC
4333		00670	BITS 2
4333	E5		PUSH HL
4334	C5		PUSH BC
4335	AF		XOR A
4336	CB1C		RR H
4338	CB1D		RR L
433A	301B		JR NC, P0000
433C	0603		LD B, 3
433E	AF		XOR A
433F	CB1C	A0000	RR H
4341	CB1D		RR L
4343	1F		RRA

4344 10F9		DJNZ	A0000
4346 07		RLCA	
4347 07		RLCA	
4348 07		RLCA	
4349 47		LD	B,A
434A 04		INC	B
434B AF		XOR	A
434C 37		SCF	
434D 17	B0000	RLA	
434E 10FD		DJNZ	B0000
4350 01D443		LD	BC, BASE
4353 09		ADD	HL, BC
		COND	Z=2
4354 2F		CPL	
		ENDC	
4355 A6		AND	(HL)
		COND	Z=2
4356 77		LD	(HL), A
		ENDC	
4357 C1	P0000	POP	BC
4358 E1		POP	HL
		ENDM	
4359 79	00680	LD	A,C
435A 85	00690	ADD	A,L
435B 6F	00700	LD	L,A
435C 78	00710	LD	A,B
435D 8C	00720	ADC	A,H
435E 67	00730	LD	H,A
435F 30D2	00740	JR	NC,L90
4361 110200	00750 L120	LD	DE,Z
4364 60	00760	LD	H,B
4365 69	00770	LD	L,C
4366 19	00780	ADD	HL,DE
4367 110001	00790	LD	DE,256
436A CD901C	00800	CALL	1C90H
436D 3BBC	00810	JR	C,L70
436F 210200	00820	LD	HL,2
4372 E5	00830	PUSH	HL
4373 CDAF0F	00840	CALL	0FAFH
4376 E1	00850	POP	HL
4377 23	00860	INC	HL
4378 CDAF43	00870 L170	CALL	TEST
437B 2817	00880	JR	Z,NO
437D 3A2040	00890	LD	A,(4020H)
4380 E607	00900	AND	7
4382 EE07	00910	XOR	7
4384 C6C1	00920	ADD	A,0C1H
4386 CD2A03	00930	CALL	32AH
4389 E5	00940	PUSH	HL
438A CDAF0F	00950	CALL	0FAFH
438D E1	00960	POP	HL
438E 3A8038	00970 WAIT	LD	A,(3880H)
4391 B7	00980	OR	A
4392 20FA	00990	JR	NZ,WAIT
4394 23	01000 NO	INC	HL
4395 23	01010	INC	HL
4396 110100	01020	LD	DE,1
4399 CD901C	01030	CALL	1C90H
439C 20DA	01040	JR	NZ,L170
439E 3E0D	01050	LD	A,0DH
43A0 CD3A03	01060	CALL	33AH
43A3 C3CC06	01070	JP	6CCH
43A6 50	01080 MESS	DEFM	'PRINTER?'
43A7 52			
43A8 49			
43A9 4E			
43AA 54			
43AB 45			
43AC 52			
43AD 3F			
43AE 00	01090	DEFB	0
43AF	01100 TEST	BITS	1
43AF E5		PUSH	HL
43B0 CS		PUSH	BC
43B1 AF		XOR	A

43B2 CB1C		RR	H
43B4 CB1D		RR	L
43B6 3019		JR	NC, P0001
43B8 0603		LD	B,3
43BA AF		XOR	A
43BB CB1C	A0001	RR	H
43BD CB1D		RR	L
43BF 1F		RRA	
43C0 10F9		DJNZ	A0001
43C2 07		RLCA	
43C3 07		RLCA	
43C4 07		RLCA	
43C5 47		LD	B,A
43C6 04		INC	B
43C7 AF		XOR	A
43C8 37		SCF	
43C9 17	B0001	RLA	
43CA 10FD		DJNZ	B0001
43CC 01D443		LD	BC, BASE
43CF 09		ADD	HL, BC
		COND	1=2
		CPL	
		ENDC	
43D0 A6		AND	(HL)
		COND	1=2
		LD	(HL), A
		ENDC	
43D1 C1	P0001	POP	BC
43D2 E1		POP	HL
		ENDM	
43D3 C9	0111	RET	
43D4	011.	EQU	\$
4300	01130	END	START
<b>00000 TOTAL ERRORS</b>			
A0000	433F		
A0001	43B8		
B0000	434D		
B0001	43C9		
BASE	43D4		
BITS	M		
G0	4312		
G02	4316		
L120	4361		
L170	4378		
L70	432B		
L90	4333		
MESS	43A6		
NO	4394		
P0000	4357		
P0001	43D1		
START	4300		
TEST	43AF		
WAIT	438E		

\*\*\*\*\* CHORD PRACTICE LII/4K

(C) DR. A.F.J. BELL \*\*\*\*\*

## INTRODUCTION.

This program displays a series of random chord names either on the screen or on a printer, letting the user practise his chord fingering either by trying to keep up with the screen display or by reading a computer printout. The program will request the user to select the KEY (musical note keyboard) in which he wishes to practise his chords, and will then proceed to display the chord names on the screen until a timing loop in the program returns you to the menu. For anybody unfamiliar with chords, here is a brief description:

A chord is a group of notes (3 or more) which are played simultaneously. They are usually (on a keyboard instrument) played with the left hand whilst the right hand plays the melody, (the recognisable portion of a tune). A knowledge of music is required to use this program, (actual chord fingerings are not displayed, only a prompt as to which chord is required). Intending users who are unfamiliar with chords and wish to learn may obtain a book on chord fingering from most music stores.

## \*\* Sample Printout from program \*\*

## Chords in G and F

C7 C G C	C7 Dm D7 Bb	Bb Dm C Bb	G F F D7
F Em Bb Dm	F G Em Bb	Dm C7 Bb G	G D7 C7 C
G G Em G	C C F Bb	F Em G Em	C7 D7 C7 C7
C7 D7 C7 Bb	G Bb Em Dm	G F Em Em	F C Bb D7
Em Bb Bb Em	F Em Bb C7	Dm Bb D7 F	Bb Bb Em Bb
C7 C7 Em D7	D7 D7 F G	Bb C7 C C	Em G Dm D7
C G D7 Em	G G D7 Dm	F Bb Dm C7	Dm Em Em Bb
G C7 C C	G Bb F D7	F Dm Bb Dm	Dm D7 Dm Dm
C C D7 C7	Em Bb C C	C7 F G Bb	Em C7 C Em
Em Dm Bb F	F C7 D7 Eb	G C7 C7 D7	C F Dm C7
D7 Em Dm F	C7 Bb C Dm	D7 Em F F	F F C7 Dm
Em D7 D7 D7	Dm D7 D7 Em	F D7 Dm C	D7 G F C7
C7 Em C7 Bb	Bb G G C	F Dm F C	F Bb Dm C7
C Bb F C	G Dm Bb G	D7 C7 F G	C7 C7 C C7
C C Dm C7	Dm F Em Bb	Dm Bb Em Bb	D7 F G G
F Bb C G	G C7 C C7	Dm Em Dm Em	Dm Em Dm Bb
G C7 Em Em	F G Em C7	Em C7 F C	Dm Dm Dm F
Bb Bb Em D7	Bb C7 Bb G	Dm Em C D7	C G G Bb
Dm Bb D7 Bb	Bm C G Dm	D7 Em Bb Em	Dm C7 Dm Dm
C7 Em Bb Dm	C7 C7 D7 Bb	G F F C	C7 C Em F
D7 Dm Em C	Em G C C7	C7 F C Em	C Dm Em Dm
F C C7 C	Bb F C7 F	C G Em D7	Em F Dm Dm
Em F C D7	D7 Bb C G	G Em D7 C7	C7 C C C
F C7 D7 Dm	F Bb Em F	C7 C Em D7	G D7 Dm Em
Bb Dm D7 F	F C G G	D7 Dm Em Bb	D7 Bb Bb F
C Dm F Dm	C7 G C7 Em	G G Em C7	C Em F C
C C7 F G	Em Bb G F	Dm Bb C F	Dm Em C7 D7
F Dm Dm D7	C7 C Em C7	C G Em D7	Em C D7 G
D7 C C Bb	Dm C7 C C	Em Em C F	C7 G D7 D7
Bb D7 Bb F	Dm Dm Dm C	C C C D7	Dm Bb Bb G

```

10 ' CHORD PRACTICE
20 ' A FJ BELL 17/03/81
30 ' Uses 2.1 Kbytes
40 DEFINT A-Z:DIM CHRD$(12),OPTN$(8)
50 ' Defines chords and options
60 FOR L=1 TO 12:READ CHRD$(L):NEXT
70 DATA C,G7,F,Am
80 DATA G,D7,C,Em
90 DATA F,C7,Bb,Dm
100 FOR L=1 TO 8:READ OPTN$(L):NEXT
110 DATA "Chords in C"
120 DATA "Chords in G"
130 DATA "Chords in F"
140 DATA "Chords in C and G"
150 DATA "Chords in C and F"
160 DATA "Chords in G and F"
170 DATA "Chords in C, G, and F"
180 DATA "End"
190 ' Displays options
200 CLS:PRINT:PRINT:PRINT CHR$(23) TAB(12)"Options":PRINT
210 FOR L=1 TO 8
220 PRINT TAB(5) CHR$(64+L);";OPTN$(L)
230 NEXT
240 PRINT TAB(7)"What is your choice ? ";
250 O$=INKEY$:IF O$=""GOTO 250
260 PRINT O$:OPTN=ASC(O$)-64
270 IF OPTN<1 OR OPTN>8 GOTO 250
280 IF OPTN=8 GOTO 570
290 FLAG=0
300 PRINT TAB(7)"Lineprinter or VDU ? "
310 R$=INKEY$:IF R$=""GOTO 310
320 IF R$="V" THEN CLS:PRINT CHR$(23):GOTO 360

```

```

330 IF R$<>"L" GOTO 310
340 IF PEEK(14312)<>63 THEN PRINT"Lineprinter not ready"
350 LPRINT CHR$(27);CHR$(66)
360 ' Sets parameters
370 IF OPTN=1 THEN RANGE= 4:FACTR=0:GOTO 440
380 IF OPTN=2 THEN RANGE= 4:FACTR=4:GOTO 440
390 IF OPTN=3 THEN RANGE= 4:FACTR=8:GOTO 440
400 IF OPTN=4 THEN RANGE= 8:FACTR=0:GOTO 440
410 IF OPTN=5 THEN RANGE=12:FACTR=0:FLAG=1:GOTO 440
420 IF OPTN=6 THEN RANGE= 8:FACTR=4:GOTO 440
430 IF OPTN=7 THEN RANGE=12:FACTR=0:GOTO 440
440 ' Displays or printsout chord
450 IF R$="L" THEN LPRINT TAB(32-LEN(OPTN$(OPTN))/2) OPTN$(OPTN)
:LPRINT" "
460 FOR L=1 TO 480
470 X=RND(RANGE)+FACTR
480 IF FLAG=1 AND (X>4 AND X<9) GOTO 470
490 IF R$="V" THEN PRINT @ 480,CHR$(X);:FOR J=1 TO 500:NEXT J:
PRINT @ 480,STRING$(2,32):FOR K=1 TO 150:NEXT K:GOTO 530
500 LPRINT USING "% %";CHR$(X);
510 IF L/4=INT(L/4) THEN LPRINT"    ";
520 IF L/16=INT(L/16) THEN LPRINT" "
530 RANDOM
540 NEXT
550 IF R$="V" GOTO 190
560 FOR L=1 TO 8:LPRINT" ":NEXT:GOTO 190
570 CLS:END

```

---

\*\*\*\*\* DOG RACE LII/4K

(C) J.D. INGRAM \*\*\*\*\*

Four dogs are lined up and move via a random number between 1 and 4 generated by the computer. The dogs are modularized and so the one subroutine will move the four dogs when the respective updated coordinates are "plugged" into the "move" subroutine.

Line 10 sets the initial coordinates for each dog. Lines 20 and 50 set the X and Y coordinates firstly to A and B and then to I and J. The GOSUB 300 is a modular "standing dog".

Lines 80 and 110 loads the selected dog's current coordinates into X and Y and then goes to the "move" subroutine. Line 150 determines that if no dog has reached the finish line, another move can be made. However, if any dog has touched the finish line then lines 160 and 190 determine which dog is the winner. So place your bets and type RUN and as the author says, "when taking my Scottie for a walk he doesn't move much faster, stopping at each blade of grass to ----".

```

10 CLS:'COPYRIGHT (C) 1978 BY J.D. INGRAM, PO BOX 244, GAWLER 5118.
20 PRINT@15,"== THIS IS A DOG RACE ==";:FORF=0TO3333:NEXTF
30 CLS:A=22:B=5:C=22:D=15:G=22:H=25:I=22:J=35
40 X=A:Y=B:GOSUB250
50 X=C:Y=D:GOSUB250
60 X=G:Y=H:GOSUB250
70 X=I:Y=J:GOSUB250
80 PRINT@252,"NO 1";:PRINT@444,"NO 2";:PRINT@636,"NO 3";:PRINT@8
28,"NO 4";
90 FORY=4TO45STEP5:SET(117,Y):NEXTY
100 PRINT@0,"";:INPUT"PRESS 'ENTER' KEY TO START RACE";A$
110 Z=RND(4)
120 IFZ=1X=A:Y=B:GOSUB290:A=X:GOTO160
130 IFZ=2X=C:Y=D:GOSUB290:C=X:GOTO160
140 IFZ=3X=G:Y=H:GOSUB290:G=X:GOTO160
150 IFZ=4X=I:Y=J:GOSUB290:I=X:GOTO160
160 IFX<120GOTO110
170 IFA>=120PRINT@128," NUMBER 1 W I N N E R ! ! ";:GOTO210
180 IFC>=120PRINT@384," NUMBER 2 W I N N E R ! ! ";:GOTO210
190 IFG>=120PRINT@576," NUMBER 3 W I N N E R ! ! ";:GOTO210
200 IFI>=120PRINT@768," NUMBER 4 W I N N E R ! ! ";:GOTO210
210 FORF=0TO2222:NEXTF:PRINT@0,"";
220 INPUT"ANOTHER RACE ? :TYPE A 1 , IF NOT TYPE A 0";S
230 IFS=1GOTO030
240 END
250 SET(X-9,Y):SET(X-20,Y):SET(X-6,Y+1):SET(X-7,Y+1)
260 SET(X-8,Y+1):SET(X-19,Y+1):SET(X-10,Y+4):SET(X-17,Y+4)
270 SET(X-11,Y+5):SET(X-16,Y+5)

```

```

280 FORU=9TO18:FORV=2TO3:SET(X-U,Y+V):NEXTU:NEXTV:RETURN
290 RESET(X-20,Y):RESET(X-19,Y+1):SET(X-17,Y+1):SET(X-16,Y)
300 SET(X-5,Y+1):SET(X-4,Y+1):RESET(X-9,Y):SET(X-6,Y)
310 RESET(X-18,Y+2):RESET(X-17,Y+2):SET(X-8,Y+2):SET(X-7,Y+2)
320 RESET(X-8,Y+1):RESET(X-7,Y+1):RESET(X-11,Y+5):RESET(X-10,Y+4)
)
330 SET(X-8,Y+4):SET(X-7,Y+5):RESET(X-18,Y+3):RESET(X-17,Y+3)
340 SET(X-8,Y+3):SET(X-7,Y+3):RESET(X-17,Y+4):SET(X-15,Y+4)
350 RESET(X-17,Y+1):SET(X-15,Y+1):RESET(X-16,Y+2):RESET(X-16,Y+3)
)
360 RESET(X-15,Y+2):RESET(X-15,Y+3):RESET(X-16,Y+5):RESET(X-15,Y+5)
370 RESET(X-15,Y+4)
380 SET(X-13,Y+4):SET(X-12,Y+5):RESET(X-8,Y+4):SET(X-6,Y+4)
390 SET(X-6,Y+2):SET(X-5,Y+2):SET(X-6,Y+3):SET(X-5,Y+3)
400 SET(X-3,Y+1):SET(X-2,Y+1):RESET(X-6,Y):SET(X-5,Y)
410 RESET(X-6,Y+1):RESET(X-5,Y+1):X=X+4:RETURN

```

\*\*\*\*\* LIST DISABLE LII/4K (C) B. HOFF \*\*\*\*\*

### INTRODUCTION.

Many has been the time when you have left a program on the '80 to do complex tasks, such as a 724 day program to prove the four-colour map theorem, only to find that on the 723rd day someone has come along, pressed BREAK and typed

```

10 PRINT "HELLO";
20 GOTO 10

```

or changed line 10360 from GOTO 560 to GOTO 560700 or something equally infuriating. Apart from looking for the correct neck to twist, did you not also wish that there was a way of stopping people from BREAKing, LISTing, EDITing or NEWing (or in some cases, CSAVEing) your programs. Well, this program is able to cut out access to the computer to the point where, short of rebooting DOS or restarting at the MEMORY SIZE? question, i.e. turning the computer off, it is impossible for any user to modify the program in memory. The comments should make operation of the program clear, but I will give two common examples. Suppose your friends insist on copying all your programs. Simply load the disable routine, add 186,-173 to line 3 and RUN. Everything will work normally except CSAVE and SAVE which will be disabled, along with BREAK. Or the computer is on display and you don't want anybody to look at the program. Load the program and extend line 3 with 185,167,174,-142 then change the data item 32 to 40. After RUNning, everything will be disabled except CLOAD,LOAD,SYSTEM and RUN. To find the keywords, run this program:-

```

10 X=128:FORI=5712T06176:PRINTCHR$(PEEK(I)AND127);:IFPEEK(I+1)
>127THENPRINTX,:X=X+1
20 NEXT

```

```

1 'LIST DISABLE (C) 1980 BY BRENTON HOFF
2 CLEAR:P=PEEK(16548)+PEEK(16549)*256:FORI=PTOP+99:READJ:POKEI,A
BS(J):IFJ>=0THENNEXTELSEK=I+1:L=I-P-34:I=P+35:POKEP+17,IAND255:PO
KEP+18,I/256:POKEP+20,LAND255:POKEP+21,L/256:FORI=0TO2:POKEP+32+I
,PEEK(16818+I):NEXT:POKE16396,241:POKE16397,201
3 POKE16819,PAND255:POKE16820,P/256:POKE16818,195:POKEK,O:K=K+1:
POKEK,O:POKEK+1,O:POKE16548,KAND255:POKE16549,K/256"

```

```

4 DATA56,24,245,229,213,229,209,19,26,183,40,17,254,128,56,247,3
3,0,0,1,0,0,237,177,32,237,195,204,6,209,225,241,201,0,0
5 '

```

HOW TO USE :  
 TYPE IN THE DECIMAL VALUES OF THE KEYWORDS YOU WANT TO  
 ENABLE/DISABLE, WITH LAST ELEMENT NEGATIVE. THEN EITHER  
 1. SIMPLY RUN IF YOU WANT ONLY THE KEYWORDS JUST TYPED IN  
 DISABLED, OR  
 6 2. CHANGE THE DATA ITEM 32 TO 40 (JR NZ, TO JR Z,)  
 FOR WHEN YOU WANT TO DISABLE ALL BUT THE KEYWORDS  
 JUST TYPED IN.

REPLACE FIRST TWO DATA ITEMS WITH ZERO IF USER WANTS  
 TO TYPE IN A BASIC PROG.

## \*\*\*\*\* SHARE TRANSACTION COSTS LII/4K (C) D. HARDIE \*\*\*\*\*

This is a short program which calculates brokerage and stamp duty costs on share transactions up to a value of \$250,000. It is accurate for all Australian States except W.A. in which stamp duty will cost a few cents more for transactions below \$100 value. Double-precision is used for the number, price and consideration variables to avoid annoying 1 and 2 cent errors in the total. Respond to the prompts in the program by entering the number of shares and then the price of the shares and the program will supply the information on the brokerage, stamp duty, buying cost and the selling proceeds.

10 ' SHARE TRANSACTION COSTS - LII/4K  
20 ' BY D. L. HARDIE

```

30 CLEAR:DEFDBLC,N,P:F$$="$$$$,###.##"
40 CLS:PRINT@64,"SHARE COSTS AND PROCEEDS CALCULATOR":PRINT$STRIN
G$(35,131):PRINT
50 INPUT"NUMBER OF SHARES      =" ;N
60 INPUT"PRICE            =" ;P:PRINT
70 C=P*N:IFC<5000THEN B=5+C*.025:GOTO130
80 IFC<15000THEN B=130+(C-5000)*.02:GOTO130
90 IFC<50000THEN B=330+(C-15000)*.015:GOTO130
100 IFC<250000THEN B=855+(C-50000)*.01:GOTO130
110 PRINT"IF YOU HAVE MORE THAN $250,000 TO INVEST THEN YOU NEED
"
120 PRINT"A PROFESSIONAL ADVISER.":GOTO200
130 IFC<100THENS=INT(C/25-.0001)*.07+.07:GOTO150
140 S=INT(C/100-.00005)*.3+.3
150 PRINT"CONSIDERATION =";:PRINTTAB(18)USINGF$;C
160 B=INT(B*100)/100:PRINT"BROKERAGE =";:PRINTTAB(18)USINGF$;B
170 PRINT"STAMP DUTY =";:PRINTTAB(18)USINGF$;S
180 PRINT:PRINT"BUYING COST =";:PRINTTAB(18)USINGF$;C+B+S
190 PRINT"SELLING PROCEEDS=";:PRINTTAB(18)USINGF$;C-B-S
200 PRINT@960,"TYPE E TO END      ANY OTHER KEY TO CONTINUE";
210 GOSUB220:IFA$$="E"THENEND ELSE 40
220 '
230 A$$=INKEY$:A$$=""
240 A$$=INKEY$:IFA$$=""THEN240ELSERETURN

```

## \*\*\*\*\* SHARE CHARTING LII/16K (C) D. HARDIE \*\*\*\*\*

This is a data handling system for day to day variations in share prices. It displays the data in graphical form on the screen. Logarithmic representation of share prices is used as equal distances on the screen represent equal percentage variation in value. The program is complicated somewhat by the need to handle different types of data, i.e. on a day to day and weekly chart, last sale prices are used, but for monthly charting a high and low range of prices is used and this complicates all of the data handling routines.

## OTHER USES.

The program could be used by a retailer to chart retail sales or virtually any other requirement for displaying percentage variations in data. The program starts by asking the user if he wishes to start a new file or load an old file. After this, the program displays the main menu which gives the following options:-

A GRAPHIC DISPLAY OF THE DATA. (displayed on a graph)

ADD DATA TO THE FILE.

EDIT AND ADD DATA TO THE FILE.

LIST THE FILE DATA.

CHANGE THE SCALE. (alter the scale of the users data, displayed on the graph)

LOAD ANOTHER COMPANY DATA AND END.

The program has been set up to operate with either a 16K cassette system or a 32K disk system for the data transfers. Simply respond to the prompts as to which type of system you have with either a "D" for disk or "T" for tape. Users not having a disk system connected should take care not to select the disk functions or the computer will lock up, making it necessary to press the reset button and, of course, losing your data at the same time. Options are also given to make back up copies of the data.

All necessary instructions and prompts are given to the user by the program itself.

```

10 '      SHARE CHARTING PROGRAM
20 '      LII/16K TAPE  (32K DISK)
30 '      BY D. L. HARDIE

40 CLEAR:CLEAR500:DEFINTD,N,X,V,Z:DIMD$(7),DM(12),DM$(12),P(280)
50 S1=43:S2=4:F1$$="#$$#:#$$":F$="#$$!##!$$#":D$="-"
60 FORN=1TO7:READD$(N):NEXT:FORN=1TO12:READDM(N),DM$(N):NEXT
70
    CLS:PRINT@77,"**** STOCK MARKET MENU ****":PRINT
80 PRINT"CREATE A NEW DATA FILE","TYPE 1"
90 PRINT"LOAD AN EXISTING FILE","TYPE 2"
100 GOSUB270:IFA$$="1"THEN120
110 IFA$$="2"THEN230ELSE70
120 '
    *** NEW DATA FILE ***
130 CLS:GOSUB570
140 N1=20:N2=20:PRINT:IFAD$$="D"THENINPUT INPUT FILESPEC (EG. BHP
/TXT)":WF$
150 INPUT"INPUT TITLE (EG. B H P)":W$
160 INPUT"INPUT IF DATA TO BE DAILY, WEEKLY OR MONTHLY (D/W/M)":
;DA$:IFDA$$="D"THENDA$$="W"THEN180ELSE170
170 IFDA$$="M"THEN200ELSE160
180 PRINT"INPUT DATE OF EARLIEST DATA":PRINTTAB(10);"DAY OF WEE
K (EG SUN=1, MON=2, ..., SAT=7)":INPUTD4
190 PRINTTAB(10);"DAY (1 TO 31)":INPUTD1
200 PRINTTAB(10);"MONTH (1-12)":INPUTD2
210 PRINTTAB(10);"YEAR (EG. 80)":INPUTD3
220 GOSUB890:GOSUB1680:GOSUB1080:GOT0930
230 '
    *** LOAD AN EXISTING FILE ***
240 CLS:K1=0:GOSUB570:IFAD$$="T"THEN260
250 CLS:INPUT "FILESPEC":WF$
260 GOSUB770:GOSUB890:GOT0930
270 '
    *** HOLD ROUTINE ***
280 A$$=INKEY$$:A$$=""
290 A$$=INKEY$$:IFA$$=""THEN290 ELSE RETURN
300 '
    *** DATE DATA ***
310 DD=D1:DM=D2:DY=D3:DN=D4:RETURN
320 '
    *** DATE INCREMENT ***
330 IFDA$$="M"THEN410
340 DD=DD+1:DN=DN+1:IFDN=8THENDN=1
350 IFDD>DM(DM)THENGOSUB390
360 IFDA$$="W"THENIFDN<>D4THEN340
370 IFDA$$="D"THENIFDN=1ORDN=2THEN340
380 RETURN
390 IFDD=DM(DM)+2THENDD=1:GOT0410
400 DD=1:IFDM=2ANDINT(DY/4)=DY/4THENDD=DM(DM)+1:GOT0360
410 DM=DM+1:IFDM=13THENDM=1:DY=DY+1
420 RETURN
430 '
    *** DATE DECREMENT ***
440 IFDA$$="M"THEN500
450 DD=DD-1:DN=DN-1:IFDN=0THENDN=7
460 IFDD=0THENGOSUB500

```

```

470 IFDA$="W"THENIFDN<>D4THEN450
480 IFDA$="D"THENIFDN=10RDN=2THEN450
490 RETURN
500 DM=DM-1:IFDM=OTHENDM=12:DY=DY-1
510 DD=DM(DM)
520 IFDM=2ANDINT(DY/4)=DY/4THENDD=DM(DM)+1
530 RETURN
540 '
*** DATE PRINT OUT ***
550 IFDA$="M"THENPRINTDM$(DM);1900+DY,:RETURN
560 PRINTD$(DN);":PRINTUSINGF$,DD,D$,DM,D$,DY,:RETURN
570 '
*** TAPE/DISK QUESTION ***
580 PRINT:PRINT:PRINT"TAPE OR DISK (T/D) ?":GOSUB270:AD$=A$
590 IFA$="T"ORA$="D"THENRETURNELSE580
600 '
*** OUTPUT ***
610 IFK1=0THENRETURN
620 IFN2>269THENNN2=269
630 CLS:GOSUB570
640 IFAD$="T"THEN690
650 IFWF$=""THENINPUT"FILESPEC";WF$
660 OPEN"O",1,WF$
670 PRINT#1,D1,D2,D3,D4,P1,P2,P3,N2,W$; ",;DA$
680 FORN=N1TON2:PRINT#1,P(N):NEXT:CLOSE:RETURN
690 PRINT@128,"PRESS PLAY & RECORD BUTTONS ON CASSETTE
WHEN READY HIT ANY KEY TO CONTINUE":GOSUB270
700 PRINT#-1,W$,D1,D2,D3,D4,P1,P2,P3,N2,DA$
710 FORN=N1TON2+1STEP16
720 PRINT#-1,P(N),P(N+1),P(N+2),P(N+3),P(N+4),P(N+5),P(N+6),P(N+
7),P(N+8),P(N+9),P(N+10),P(N+11),P(N+12),P(N+13),P(N+14),P(N+15)
730 NEXT
740 PRINT"DO YOU WANT TO RECORD THE DATA AGAIN ? (Y/N)":GOSUB270
750 IFA$="N"THENRETURN
760 IFA$="Y"THEN690ELSE740
770 '
*** INPUT ***
780 IFAD$="T"THEN830
790 OPEN"I",1,WF$
800 N1=20:INPUT#1,D1,D2,D3,D4,P1,P2,P3,N2,W$,DA$
810 FORN=N1TON2:INPUT#1,P(N):NEXT
820 CLOSE:RETURN
830 CLS:PRINT@128,"PRESS PLAY BUTTON ON CASSETTE
WHEN READY HIT ANY KEY TO CONTINUE":GOSUB270
840 INPUT#-1,W$,D1,D2,D3,D4,P1,P2,P3,N2,DA$
850 N1=20:FORN=N1TON2+1STEP16
860 INPUT#-1,P(N),P(N+1),P(N+2),P(N+3),P(N+4),P(N+5),P(N+6),P(N+
7),P(N+8),P(N+9),P(N+10),P(N+11),P(N+12),P(N+13),P(N+14),P(N+15)
870 NEXT:RETURN
880 '
*** INITIAL DATA ***
890 IFDA$="M"THENWH$="HIGH":WL$="LOW":W1$="MONTHLY":V1=6:V2=6
O:V3=6
900 IFDA$="D"THENWH$="":WL$=WH$:W1$="DAILY":W2$="FOUR WEEK
DIVISIONS":V1=3:V2=63:V3=10
910 IFDA$="W"THENWH$="":WL$=WH$:W1$="WEEKLY":W2$="SIX MONTH
DIVISIONS":V1=11:V2=63:V3=13
920 RETURN
930 '
*** MAIN MENU ***
940 CLS:PRINT@74,"*** MENU - ";W$;" ";W1$;" ***":PRINTTAB(10)ST
RING$(40,131)
950 PRINT"GRAPHIC DISPLAY",,"TYPE 1"
960 PRINT"TO ADD NEW DATA",,"TYPE 2"
970 PRINT"TO EDIT AND ADD DATA",,"TYPE 3"
980 PRINT"TO LIST DATA",,"TYPE 4"
990 PRINT"TO CHANGE THE SCALE",,"TYPE 5"
1000 PRINT"LOAD ANOTHER COMPANY DATA",,"TYPE 6"
1010 PRINT"SAVE DATA AND END",,"TYPE 7"
1020 GOSUB270
1030 A=VAL(A$):ONAGOSUB1440,1080,1190,1370,1680,600,620
1040 IFA=7THENEND
1050 IFA=6THEN40
1060 GOT0930
1070 RETURN
1080

```

```

*** NEW DATA ***
1090 CLS:K1=1:N=N2:GOSUB300:IFP(N2)=OTHEN1120
1100 PRINT"LAST ENTRY WAS ";TAB(26);:GOSUB540:PRINTWH$;"PRICE ="
";P(N):IFDA$="M"THENPRINTTAB(35);WL$;"PRICE =";P(N-1)
1110 GOTO1140
1120 PRINT"TO EXIT TYPE O ";:GOSUB540:P(N)=O:PRINTWL$"
;"PRICE =";:INPUTP(N):IFP(N)=OTHEN1150
1130 IFDA$="M"THENPRINTTAB(35);WH$;"PRICE =";:INPUTP(N+1):IFP(N+
1)=OTHEN1130ELSE=N+1
1140 N=N+1:GOSUB320:GOTO1120
1150 N=N-1
1160 GOSUB430:N2=N:D1=DD:D2=DM:D3=DY:D4=DN
1170 IFDA$<>"M"THENIFN2>147THENNN3=N2-147:FORN=N1TO147:P(N)=P(N+N
3):NEXT:N2=147
1180 RETURN
1190 '
*** EDIT AND ADD DATA AT END OF LIST ***
1200 CLS:N=N2:K1=1:T$=WH$:GOSUB300:PRINT"TO EXIT ENTER O"
1210 IFT$=WH$THENGOSUB540
1220 PRINTTAB(9);T$;"P("N") ="P(N) ;"    NEW VAL OR <EN> IF OK";:T
=-1:INPUTT
1230 IFT=-1THEN1260
1240 IFT=OTHEN1320
1250 P(N)=T
1260 IFP(N)=OTHEN1320
1270 IFDA$="M"THENIFT$=WH$THENSET$=WL$:GOTO1290:ELSET$=WH$
1280 GOSUB430
1290 N=N-1
1300 IFN=2THEN1320
1310 GOTO1210
1320 N=N+1
1330 IFN>=N1THENRETURN
1340 IFT$<>WH$THENNN=N+1
1350 N3=20-N:N2=N2+N3:FORN=N2TON1STEP-1:P(N)=P(N-N3):NEXT
1360 FORN=1TO19:P(N)=0:NEXT:GOTO1170
1370 '
*** LIST DATA ***
1380 GOSUB300:CLS:PRINT:N=N2
1390 GOSUB540:PRINTWH$;"P("N") ="P(N)
1400 IFDA$="M"THENNN=N-1:PRINTTAB(9);WL$;"P("N") ="P(N)
1410 IFN=N1THEN1430
1420 N=N-1:GOSUB430:GOTO1390
1430 GOSUB270:RETURN
1440 '
*** GRAPHIC DISPLAY ***
1450 CLS:F2=(S1-S2)/(LOG(P2)-LOG(P1)):F1=S1+F2*LOG(P1)
1460 FORX=15360TO16320STEP64:POKEX,191:NEXT:PRINT@961,STRING$(62
,176);
1470 FORV=V1TOV2STEPV3:FORN=0TO14:PRINT@N*64+V,".";:NEXTN,V
1480 FORP=P1TOP2STEPP3:Z=F1-F2*LOG(P):FORN=0TO127STEP6:SET(N,Z):
NEXTN:PRINT@INT(Z/3)*64+1,USINGF1$;P,:NEXTP
1490 PRINT@975,W2$;:PRINT@B,W$;"    ";W1$;" PRICE";:GOSUB300:PRIN
T"    ";:GOSUB540:PRINT@80,(P2-P1)*100/P1;"% SCALE";
1500 IFDA$="M"THEN1600
1510 X=126:N=N2:ZT=F1-F2*LOG(P(N))
1520 Z=F1-F2*LOG(P(N)):IFZ>470RZ<OTHENGOSUB1590:RETURN
1530 IFZ>ZT:FORZ1=ZT+1TOZ:SET(X,Z1):NEXT:GOTO1560
1540 IFZ<ZT:FORZ1=ZTOZT-1:SET(X,Z1):NEXT:GOTO1560
1550 SET(X,Z)
1560 ZT=Z:X=X-1:IFX=OTHENGOSUB270:RETURN
1570 N=N-1:IFN<N1THENGOSUB270:RETURN
1580 GOTO1520
1590 GOSUB270:CLS:PRINT@448,"SCALE OVEFLOW - CHANGE SCALE":GOSUB
270:RETURN
1600 Y=0:FORX=1014TO966STEP-6:PRINT@X,1900+D3-Y;:Y=Y+1:NEXT:X=10
8+D2
1610 N=N2
1620 Z1=F1-F2*LOG(P(N)):IFZ1>470RZ1<OTHENGOSUB1590:RETURN
1630 Z2=F1-F2*LOG(P(N-1)):IFZ2>470RZ2<OTHENGOSUB1590:RETURN
1640 FORZ=Z1TOZ2:SET(X,Z):NEXTZ
1650 X=X-1:IFX=OTHEN1670
1660 N=N-2:IFN>19THEN1620
1670 GOSUB270:RETURN
1680 '

```

```
**** SCALE CHANGING ROUTINE ****
1690 CLS:K1=1:PRINT@64,"GRAPH SCALE SELECTION AND CHANGING ROUTI-
NE":PRINTSTRING$(40,131)
1700 PRINTP1;TAB(20);"SELECT BASE PRICE:";:T=0:INPUTT:IFT<>OTHEN
P1=T
1710 IFP1=OTHENPRINT" MUST BE A POSITIVE NUMBER":GOTO1700
1720 PRINTP2;TAB(20);"SELECT TOP PRICE:";:T=0:INPUTT:IFT<>OTHENP
2=T
1730 PRINTP3;TAB(20);"SELECT DIVISION SIZE";:T=0:INPUTT:IFT<>OTH
ENP3=T
1740 RETURN
1750
    DATASUN,MON,TUE,WED,THU,FRI,SAT
1760 DATA31,JAN,28,FEB,31,MAR,30,APR,31,MAY,30,JUN,31,JUL,31,AUG
,30,SEP,31,OCT,30,NOV,31,DEC
```

\*\*\*\*\* PSYCHIC MASTER LII/16K (C) G. EGEL \*\*\*\*\*

Psychic master was written to test the abilities of a person claiming to have psychic abilities. The program has three options (1) HUNCH, (2) TELEPATHY, (3) CLAIRVOYANCE.

#### (1) HUNCH

Selecting HUNCH enables a person to test his ability to predict a three digit number. The odds against doing this correctly are one thousand to one. The odds against predicting a two digit number correctly are fifty to one, and against predicting correctly a single digit, three to one.

#### (2) TELEPATHY TEST

Five different symbols are displayed on the screen in random order. One person must look at the screen and try to transmit the image displayed to the test subject who, of course, should not be in a position to see the screen. When the person being tested thinks he knows the symbol, he should say the name of the symbol he thinks he has received, and the person doing the transmitting should type in the test subject's response. The program provides 25 attempts then displays the results and announces if the results fall within the bounds of chance or if the person is likely to be telepathic.

#### (3) CLAIRVOYANCE

In this test the subject has to perceive the image that the computer would have displayed if it had been seen, and type in what they think the computer would have displayed. At the end of this test, results similar to the above test are displayed.

```
10 REM## PSYCHIC MASTER ###
20 REM ## COPYRIGHT GEOFFREY EGEL MARCH 1980 ##
30 REM # 18 STURT ST LOXTON 5333 #
40 RANDOM:CLEAR 100:DIM B(25):DEFINT N,S,T
50 CLS:PRINT TAB(25)"PSYCHIC MASTER":PRINT STRING$(64,191)
60 PRINT" (1) PSYCHIC NUMBER HUNCH TRAINER":PRINT " (2) CLAIRVOY-
ANCE AND TELEPATHY TESTS": PRINT STRING$(64,140 )
70 A$=INKEY$:IF A$="" THEN 80ELSE IF A$="1" THEN 90ELSE IF A$="2"
" THEN 320
80 FOR N=1 TO 63: PRINT@576,TAB(N)"EXTRA SENSORY PERCEPTION":NEX-
T N: GOTO 70
90 CLS:PRINT " PSYCHIC NUMBER GUESsing TRAINER":PRINT STRING$(64
,191): PRINT"INSTRUCTIONS Y/N"
100 A$=INKEY$:IF A$="" THEN 100ELSE IF A$="N" THEN 140
110 CLS: PRINT"THE AIM OF THIS PROGRAM IS TO TRAIN THE MIND TO P-
RECALL THE THREE DIGIT TARGET NUMBER , TO PLAY ENTER A THREE
DIGIT NUMBER BETWEEN 000 - 999 "
120 PRINT"A HUNDRED TRIES PER DAY IS REQUIRED FOR EIGHT WEEKS TO
ACHIEVE A GOOD RESULT AND AFTER THIS YOU MUST CONTINUE TO PRACT-
ICE TO REMAIN PROFICIENT
130 INPUT"PRESS ENTER TO CONTINUE":A$
140 CLS
150 RANDOM:FOR T=1 TO 100
160 K=0
170 A(1)=RND(10)-1:A(2)=RND(10)-1:A(3)=RND(10)-1
180 A=A(1):B=A(2):C=A(3):T$=STR$(A)+STR$(B)+STR$(C):CLS:PRINT"PL-
ACE A THREE DIGIT NUMBER HERE PLEASE"
190 X$=INKEY$:IF X$="" THEN 190ELSE A(4)=VAL(X$):PRINT X$;
```

```

200 Y$=INKEY$: IF Y$="" THEN 200ELSE A(5)=VAL(Y$):PRINT Y$;
210 Z$=INKEY$: IF Z$="" THEN 210ELSE A(6)=VAL(Z$):PRINT Z$;
220 FOR N=1 TO 3:FOR S=4 TO 6
230 IF A(N)=A(S) THEN A(S)=20:A(N)=19: K=K+1
240 NEXT S:NEXT N
250 IF K=3 THEN F=F+1
260 IF K=2 THEN G=G+1
270 IF K=1 THEN H=H+1
280 CLS:P$=X$+Y$+Z$:PRINT"TARGET ";T;" WAS ";T$;" YOUR CHOICE
";P$:FOR Q=1 TO 1000: NEXT Q
290 NEXT T
300 CLS:PRINT F;"CORRECT THREE DIGITS":PRINT G;"CORRECT TWO DI
GITS":PRINT H;"CORRECT ONE DIGITS"
310 END
320 CLS
330 PRINT" GRAPHICS TEST FOR CLAIRVOYANCE AND TELEPATHY":PRINT S
TRING$(64,191):PRINT"INSTRUCTIONS Y/N"
340 A$=INKEY$: IF A$="Y" THEN 350ELSE IF A$="N" THEN 390ELSE 340
350 PRINT@192," THIS SECTION CAN BE PLAYED IN ONE OF TWO WAYS E
ITHER AS A SELF TEST CLAIRVOYANCE WHERE YOU WILL NOT GET TO SEE T
HE GRAPHICS OR AS A TEST FOR TELEPATHY WITH TWO PERSONS"
360 PRINT"ONE SEEING THE IMAGE AND TRANSMITTING IT THE OTHER NOT
SEEING AND TRYING TO PERCIEVE IT // THE FOLLOWING ARE THE S
YMBOLS":FOR Q=1 TO 4500: NEXT Q
370 Z=Z+1: ON Z GOTO 570,580,590,600,620,390
380 FOR Q=1 TO 1000:NEXT Q:GOTO 370
390 Z=0:PRINT "CLAIRVOYANCE OR TELEPATHY C/T":T=0
400 N$=INKEY$: IF N$="C" OR N$="T" THEN 410ELSE 400
410 FOR N=1 TO 4:CLS:PRINT"SHUFFLING WON'T BE LONG":FOR T=1 TO 2
5 :B(T)=0: NEXT T:FOR F=1 TO 5:FOR A=1 TO 5
420 S=RND(25):IF B(S)=0 THEN B(S)=A:GOTO 440
430 GOTO 420
440 NEXT A:NEXT F:FOR T=1 TO 25:CLS:IF N$="T" THEN ON B(T) GOTOS
70,580,590,600,620:ELSE CLS:PRINT "CLAIRVOYANCE TARGET";T
450 GOSUB 540
460 NEXT T: PRINT"END OF TEST ";N:FOR Q=1 TO 500:NEXT Q:NEXT N
470 CLS:PRINT " E.S.P RATING SCALE":PRINT STRING$(63,140)
480 PRINT"# TESTS EXP RESULT SLIGHT SIGNIFICANT HIGH E.S.P":
PRINT STRING$(63,46)
490 PRINT" 4      20      21-27      28-31      32 OR MORE
500 PRINT" 10      50      51-62      63-68      69 OR MORE
510 PRINT" 50      250     251-278     279-292     293 OR MORE
520 PRINT"100      500      501-539      540-559      560 OR MORE
530 :PRINT STRING$(64,140):PRINT "SUBJECTS RATING FOR FOUR TESTS
WAS";P: END
540 PRINT"SUBJECTS CHOICE":PRINT"(1) RECTANGLE (2) CROSS (3) LIN
ES (4) DIAMOND (5) PYRAMID"
550 A$=INKEY$: IF A$="" THEN 550ELSE:E=VAL(A$):IF (E>5)OR(E=0)THE
N 550ELSE IF E=B(T)THEN P=P+1
560 RETURN
570 CLS:PRINT TAB(23)"TELEPATHY TARGET";T:FOR K=1 TO 9: PRINT T
AB(14) STRING$(30,191):NEXT K:PRINT:PRINT TAB(23)"RECTANGLE":IF Z
=0 THEN 450ELSE 380
580 CLS: PRINT TAB(23) "TELEPATHY TARGET";T:PRINT:PRINT:FOR K=1
TO 3: PRINT TAB(23)STRING$(9,191):NEXT K:FOR K=1 TO 3:PRINT TAB(1
4) STRING$(27,191):NEXT K:FOR K=1 TO 3 :PRINT TAB(23) STRING$(9,1
91):NEXT K :PRINT TAB(23)"CROSS": IF Z=0 THEN 450ELSE 380
590 CLS:PRINT TAB(23)"TELEPATHY TARGET";T:PRINT:FOR K=1 TO 5:PRI
NT TAB(14)STRING$(26,191):PRINT:NEXT K:PRINT TAB(23)"LINES":IF Z=
0 THEN 450ELSE 380
600 CLS:PRINT TAB(23)"TELEPATHY TARGET";T:PRINT:PRINT TAB(26)STR
ING$(3,191):PRINT TAB(23) STRING$(8,191):PRINT TAB(21) STRING$(13
,191):PRINT TAB(18) STRING$(18,191)
610 PRINT TAB(16)STRING$(22,191):PRINT TAB(18) STRING$(18,191):P
RINT TAB(21)STRING$(13,191):PRINT TAB(23)STRING$(8,191):PRINT TAB
(26)STRING$(3,191):PRINT:PRINT:PRINT TAB(23)"DIAMOND":IF Z=0 THEN
450ELSE 380
620 CLS:PRINT TAB(23) "TELEPATHY TARGET";T:PRINT:PRINT:PRINT TAB
(27) STRING$(1,191):PRINT TAB(26) STRING$(3,191):PRINT TAB(24)STR
ING$(7,191):PRINT TAB(22) STRING$(12,191):PRINT TAB(20) STRING$(1
5,191)
630 PRINT TAB(19)STRING$(17,191):PRINT TAB(17)STRING$(21,191):PR
INT TAB(16)STRING$(23,191):PRINT :PRINT TAB(23)"PYRAMID":IF Z=0
THEN 450ELSE 380

```

## \*\*\*\*\* NEXT MONTH'S ISSUE \*\*\*\*\*

Next month's issue will contain at least the following programs plus the usual features and articles.

## \*\* SHARE CHARTING LI/4K \*\*

Just so you Level I folks won't feel left out of all this money-making that's going on after this month's Level 2 share market programs, next month we will have one just for you. Share charting draws a graph to display percentage rises and falls in share prices over a given period.

## \*\* BLOW FLY LII/16K \*\*

A really funny two-player game. Each player has to crawl the fly up the window without getting swatted or eaten by the spider. Complete with graphics.

## \*\* BINGO LII/4K \*\*

This program has actually been used in a club as a bingo caller. It can be used in two modes, as a two-player game of bingo or as an automatic bingo caller. It functions just like the real thing, CLICKETY CLICK.

## \*\* GENIUS LII/16K \*\*

This program is the one you have been waiting for. It will finally silence those friends of yours who are always complaining that your computer games are too easy. The only problem you will have now though is getting them to go home again.

## \*\* STARSHOOT LII/4K \*\*

This is a computer version of the very popular RUBIKS CUBE and if you have ever played with a Rubiks cube you just know right now that this program is going to drive you clear up the wall.

**APPLICATION FOR PUBLICATION  
OF A PROGRAM  
IN MICRO-80**

Date .....

Tick where appropriate

**To MICRO-80**  
*Please consider the enclosed program for . . .*

- (i) Publication in MICRO-80 .....
- (ii) Publication on disk or cassette only .....
- (iii) Both .....

Name .....

Address .....

Postcode .....

## \*\*\* CHECK LIST \*\*\*

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padded bags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

\*\*\*\*\* CASSETTE EDITION INDEX \*\*\*\*\*

The cassette edition of MICRO-80 contains all the software listed each month, on cassette. All cassette subscribers need do is CLOAD and RUN the programs. Level II programs are recorded on side 1 of the cassette. Level I programs are recorded on side 2. Level I programs are not compatible with the System 80. All programs are recorded twice in succession. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

<u>SIDE ONE</u>	<u>TYPE</u>	<u>I.D.</u>	<u>APPROX.</u>	<u>START</u>	<u>POSITION</u>
			<u>CTR-41</u>	<u>CTR-80</u>	<u>SYSTEM</u> <u>80</u>
DOG RACE	L2/4K	D	7	5	5
"	"		25	17	17
CHORD PRACTISE	L2/4K	C	43	29	29
"	"		60	41	41
SHARE TRANSACTION COSTS	L2/4K	S	77	52	52
"	"		87	59	59
SHARE CHARTING	L2/16K	A	100	67	67
"	"		148	100	100
PSYCHIC MASTER	L2/16K	P	194	131	131
"	"		220	150	150
PRIME GENERATOR	SYSTEM	PRYME	250	169	169
"	"		255	172	172
"	EDTASM	PRYME	260	175	175
"	"	"	272	184	184
LIST DISABLE	L2/4K	L	287	194	194
"	"	"	296	200	200
<u>SIDE TWO</u>					
INTERCEPT	L1/4K	-	12	8	-
"	"	-	53	36	-
BREAKOUT	L1/4K	-	93	63	-
"	"	-	148	100	-

# A SPECIAL OFFER!

TO  
SUBSCRIBERS OF

# MICRO 80

FOR THE FIRST TIME IN AUSTRALIA!  
GENUINE TANDY TRS-80 MICRO-COMPUTERS,  
HARDWARE AND SOFTWARE AT

## GREAT DISCOUNT PRICES !!!

### HOW TO ORDER -

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION  
CAT. No AND A CHEQUE OR MONEY ORDER.

### WE WILL -

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS \*
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP  
YOU INFORMED OF CURRENT SPECIALS !

\* subject to availability



to CONQUEST ELECTRONICS Pty. Ltd. 212 Katoomba St. KATOOMBA 2780			X
Please supply - QTY. CAT NO DESC.			ADV. PRICE
			SUB TOTAL
			LESS 10%
			FIND CHEQUE FOR TOTAL
SEND FREIGHT FREE TO			
NAME .....			
ADDRESS .....			
P/ Code .....			

**CONQUEST ELECTRONICS**  
PTY. LTD.  
212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491

# **MICRO-80**

## **LEVEL II ROM REFERENCE MANUAL**

by Edwin Paay

Published by MICRO-80 PRODUCTS

Written by Eddy Paay, the LEVEL II ROM REFERENCE MANUAL is the most complete explanation of the Level II BASIC interpreter ever published.

Part 1 lists all the useful and usable ROM routines, describes their functions explains how to use them in your own machine language programs and notes the effect of each on the various Z 80 registers.

Part 1 also details the contents of system RAM and shows you how to intercept BASIC routines as they pass through system RAM. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory—the only restriction is your own imagination!

Part 2 gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements, etc. It also describes the various formats used for BASIC, SYSTEM and EDITOR/ASSEMBLER tapes. Each section is illustrated by sample programs which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

The LEVEL II ROM REFERENCE MANUAL is intended to be used by machine language programmers. It assumes a basic understanding of the Z 80 instruction set and some experience of Assembly Language programming. But BASIC programmers too will benefit from reading it. They will gain a much better insight into the functioning of the interpreter which should help them to write faster, more concise BASIC programs.

# **MICRO-80**